

Reconocimiento De Caracteres Usando Una Tarjeta Zynq-7000 SoC

J.I. Vega-Luna¹, G. Salgado-Guzmán¹, M.A. Lagos-Acosta¹, J.F. Cosme-Aceves¹, F.J. Sánchez-Rangel¹

Resumen—Se presenta un sistema de reconocimiento de caracteres usando una tarjeta Zynq-7000 SoC. El objetivo fue contar con un sistema capaz de reconocer letras y concatenarlas para formar palabras cortas y reproducirlas en un altavoz, utilizando archivos de voz previamente grabada. El sistema se realizó para apoyar a personas débiles visuales con dificultades para leer textos y se implantó usando un altavoz y una cámara de video conectados a una tarjeta Zynq-7000 SoC con un FPGA y un procesador ARM Cortex-A9. La metodología usada fue dividir el sistema en dos módulos que realizan las siguientes funciones: registro y reproducción de voz, captura y almacenamiento de imágenes para reconocer caracteres usando el software OpenCV. Para el reconocimiento de caracteres se utilizó la técnica de coincidencia de plantillas, la cual compara imágenes de plantillas con una imagen para determinar si existe coincidencia. Los caracteres a reconocer son capturados como cuadros de video y almacenados usando un mapa de bits. Mediante OpenCV se determina el carácter y se reproduce en el altavoz el sonido o fonograma correspondiente. El porcentaje de exactitud fue 99.6% y el tiempo de reconocimiento fue 100 ms.

Palabras claves—Coincidencia de plantillas, FPGA, OpenCV, reconocimiento de caracteres, Zynq-7000 SoC.

Abstract—A character recognition system using a Zynq-7000 SoC card is presented. The objective was to have a system able to recognize letters, concatenate them to form short words and reproduce them in a loudspeaker, using previously recorded voice files. The system was made to support visually impaired people with reading difficulties and was implemented using a loudspeaker and a video camera connected to a Zynq-7000 SoC card with an FPGA and an ARM Cortex-A9 processor. The methodology used was to divide the system into two modules that perform the following functions: voice recording and playback, capture and storage of images to recognize characters using the OpenCV software. For character recognition, the template matching technique was used, which compares images of templates with an image to determine if there is a match. The characters to be recognized are captured as video frames and stored using a bitmap. The character is determined by OpenCV and the corresponding sound or phonogram is

reproduced on the loudspeaker. The accuracy percentage was 99.6% and the recognition time was 100 ms.

Keywords—Character recognition, FPGA, matching templates, OpenCV, Zynq-7000 SoC.

I. INTRODUCCIÓN

La función visual constituye un sistema sensorial importante por el contenido de información que aporta al cerebro. El sistema sensorial se modifica dependiendo de las condiciones de contraste, color e iluminación, como la luz natural o artificial. Las personas con función visual deficiente o limitada, se les conoce como débiles visuales. Estas personas tienen una disminución de la visión de grado variable, pueden percibir luz, ver bultos, contar dedos o movimiento de manos a corta distancia, que no mejora con tratamiento médico, quirúrgico u óptico (anteojos). Esto puede representar una pérdida temporal o permanente de la vista y constituye un obstáculo para el desarrollo normal de la vida del individuo. El objetivo de este trabajo fue desarrollar e implantar un sistema para el reconocimiento de caracteres, letras y dígitos, para apoyar a los débiles visuales en la identificación de éstos reproduciendo el fonema del carácter en un altavoz, es decir, que la persona escuche el fonema del carácter presentado al sistema aquí desarrollado.

El reconocimiento óptico de caracteres (OCR-Optical Character Recognition) es la técnica que consiste en tomar imágenes o fotografías de letras o dígitos y convertirlos a información entendible por una computadora. El OCR incluye tareas de pre-procesamiento, segmentación, extracción de características, clasificación y reconocimiento. Esta técnica ha sido estudiada, desarrollada y aplicada ampliamente en los últimos años. Algunas investigaciones realizadas incorporan algoritmos, como el Zidouri, para la segmentación, el de Blob para eliminación de áreas no deseadas y extracción de caracteres y el algoritmo de Hildrich. En algunas aplicaciones el OCR extrae diferentes características de una imagen para clasificar su contenido como letras o dígitos.

Dos de los métodos más usados para realizar OCR son el entrenamiento de un clasificador en cascada para cada letra y la comparación de plantillas. El primer método usa un archivo que indica las características que deben estar

¹ Universidad Autónoma Metropolitana-Azcapotzalco, Departamento de Electrónica, Área de Sistemas Digitales, Av. San Pablo 180, Colonia Reynosa, C.P. 02200, Ciudad de México, México.

* vlji@azc.uam.mx.

presentes en una imagen para determinar que el objeto deseado se encuentra en ella. El archivo del clasificador se usa para detectar objetos en una imagen. El clasificador crea un conjunto de ejemplos entrenamiento, positivos y negativos, del objeto a identificar. Los ejemplos positivos contienen el objeto y los negativos no lo contienen. Usando la biblioteca de funciones de código abierto Open CV, se pueden generar archivos para el clasificador del tipo Haar o LBP basados en un conjunto de ejemplos proporcionados por el usuario.

El método de comparación de plantillas es más simple que el primero, ya que toma plantillas de imágenes existentes y las compara con una imagen para determinar la similitud de la plantilla con cada región de la imagen. Cada platilla es movida en toda la imagen y, dependiendo de la exactitud del modo de comparación, se calcula y registra la calidad de la comparación. La calidad de la comparación de la plantilla con cada región se indica con un coeficiente de correlación.

Los últimos avances en el desarrollo de OCR se han enfocado a la aplicación de esta técnica para reconocer caracteres y símbolos de una imagen a texto editable por una computadora de idiomas como el Malayo, Árabe, Chino y Japonés, entre otros. Se ha creado una diversidad de investigaciones debido a que en algunos idiomas el OCR es una tarea desafiante en la cual los métodos tradicionales requieren mejoras [1]. Algunas investigaciones realizan reconocimiento en línea y otras a partir de un texto previamente escrito. Otras trabajan con OCR en dispositivos móviles y otras usando una computadora. Algunas investigaciones combinan OCR con: redes neuronales [2], el modelo oculto de Markov [3], análisis discriminante lineal (LDA-Linear Discriminant Analysis-LDA) y vecinos más cercanos (NN-Nearest Neighbor) [4], vectores de máquinas y clasificadores discriminantes cuadráticos (QDA-Quadratic Discriminate Classifier-QDA) [5], árboles de decisión y códigos de percepción [6], el script de Devanagari [7] y, las más recientes, con visión por computadora [8].

En cuanto al código usado para implantar OCR, algunas investigaciones han desarrollado código propio y otras han usado algoritmos de código abierto como en el caso del motor de Tesseract [9] y el de extracción de características de proyección horizontal y vertical [10].

El OCR se aplica actualmente en diversos campos de la tecnología y vida humana como por ejemplo: en reconocimiento de documentos bancarios [11], en lectura de números de serie de motores, en reconocimiento de gesticulaciones humanas [12], en lectura de placas de vehículos [13] y en apoyo a invidentes y débiles visuales. Algunas implantaciones de OCR, usan una computadora de propósito específico con una cantidad de recursos

considerable, ya que el método y algoritmos usados así lo demandan [14]. Otras implantaciones, las que han logrado menor tiempo de respuesta, han sido implantadas en hardware, usando un FPGA y un HPS (Hard Processor System). Particularmente, las aplicaciones de reconocimiento de objetos en una secuencia de video han sido implantadas en su mayoría utilizando FPGA [15].

Después de realizar una investigación para seleccionar el método de OCR más adecuado a usar en este trabajo, se optó por utilizar el de comparación de plantillas, ya que es el más sencillo de entender e implantar [16]. Se consideró que para la aplicación a realizar no se necesita un método o algoritmo más sofisticado. A pesar de ser el más sencillo, la cantidad de recursos de cómputo que demanda el método de comparación de plantillas, el tiempo de respuesta y exactitud que se lograron son aceptables, razón por la cual se implantó a través de un FPGA en la aplicación aquí presentada [17].

Un FPGA (Field Programmable Gate Array) es un dispositivo lógico programable que incorpora gran cantidad de compuertas lógicas y circuitos digitales cuya interconexión y funcionalidad puede configurarse mediante un lenguaje de descripción de hardware o una herramienta gráfica. Esto los hace flexibles al flujo de diseño y presentan costo de adquisición bajo para pequeñas cantidades de dispositivos, además que el tiempo de desarrollo es menor en comparación con otros dispositivos digitales. Se usan cada vez más en aplicaciones de procesamiento digital de señales y sistemas electrónicos de control. Permiten implantar sistemas complejos en un circuito integrado reemplazando a tarjetas de desarrollo que contienen varios circuitos, interfaces y buses.

Uno de los fabricantes de FPGA es Xilinx. Los FPGA de Xilinx incluyen procesadores con características y funcionalidades a la medida, memoria y transceptores de alta velocidad. Un procesador que puede implantarse con la lógica programable y bloques de memoria de las familias de FPGA de Xilinx, es el procesador ARM Cortex-A9. Es uno de los procesadores embebidos más utilizados en la industria de los FPGA. Es equilibrado en rendimiento y tamaño para aplicaciones de procesamiento de información. Tiene arquitectura RISC de 32 bits y la naturaleza de su núcleo por software permite que el ARM Cortex-A9 sea creado con características específicas para cada aplicación. Puede extenderse la funcionalidad básica del ARM Cortex-A9, adicionándole unidades de manejo y protección de memoria e instrucciones y periféricos personalizados para ejecutar un sistema operativo.

El uso de un FPGA presenta tres ventajas: son dispositivos reprogramables, disminuyen el costo y tiempo de desarrollo y proporcionan mayor rendimiento a las

aplicaciones. Debido a estas ventajas, así como a la frecuencia alta de trabajo y capacidad de procesamiento en paralelo, los FPGA exceden la potencia de cómputo de los procesadores digitales de señales (DSP).

La variedad de aplicaciones de un FPGA es muy amplia, entre las cuales se encuentran: sistemas de visión, tratamiento de imágenes médicas, radio por software, reconocimiento de voz e imágenes, codificación y encriptación, radioastronomía y aeronáutica. Existen proveedores de tarjetas de desarrollo que conjuntan un FPGA con circuitos de memoria, interfaces serie y de red, interruptores y dispositivos de salida como leds y displays LCD. Una de las tarjetas de desarrollo de Xilinx es la Zync-7000 SoC, la cual fue utilizada en este trabajo. Xilinx ofrece el software Vivado para el diseño y simulación de circuitos en la tarjeta Zync-7000 SoC y soporta el uso de los lenguajes de descripción de hardware VHDL y Verilog. La programación realizada en Vivado se transfirió desde una computadora a la Zync-7000 SoC para probar el diseño de la aplicación aquí presentada. El procesador ARM Cortex-A9 implantado en el FPGA y los diferentes componentes de la Zync-7000 SoC permitieron crear el sistema digital completo. Para implantar el sistema embebido con el procesador ARM Cortex-A9, se hizo uso de las herramientas SOPC Builder y Qsys que incorpora Vivado, utilizando la interfaz gráfica para establecer las características del procesador y seleccionar los componentes como: periféricos, bloques de entrada/salida, temporizadores, interfaces y controladores, en lugar de escribir código para realizar el sistema embebido. Por estas funcionalidades y los recursos de la tarjeta Zync-7000 SoC se utilizó un FPGA para obtener un sistema práctico, confiable y eficiente en el desarrollo de la aplicación desarrollada.

La ventaja de utilizar el método de comparación de plantillas, en este sistema, es que se implanta de una manera simple, aunque requiera más hardware, además de ser el método de OCR, más usado, en investigaciones recientes. La aportación de este trabajo es la ayuda que proporciona a débiles visuales, de una manera eficiente, identificando caracteres y dígitos, a través de un sistema realizado con componentes de reciente tecnología, bajo costo y, en su mayoría, con software de código abierto.

II. PARTE TÉCNICA DEL ARTÍCULO

La metodología usada en el desarrollo de este trabajo fue dividirlo en cinco módulos: el módulo de la tarjeta Zync-7000 SoC, el módulo de entrada de video, el módulo de entrada de audio, el módulo de salida de audio y el módulo de reconocimiento. Una vez definidos los módulos del sistema se procedió a identificar las funciones específicas y componentes de cada uno de ellos usando los siguientes

principios: el sistema resultante debe ser compacto, confiable y fácil de utilizar. En la Figura 1 se muestra el diagrama de bloques del sistema. La función y componentes de cada módulo se explican a continuación.

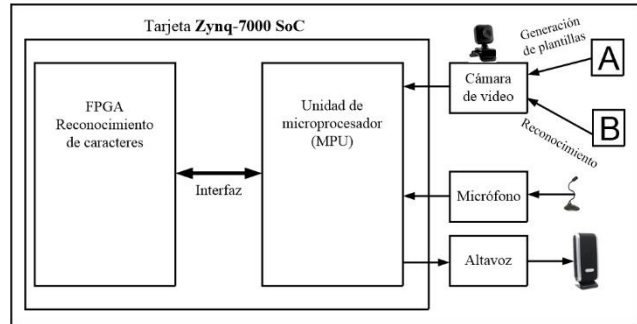


Figura 1. Bloques funcionales del sistema

A. Módulo de la tarjeta Zync-7000 SoC

La tarjeta Zync-7000 SoC tiene un hardware robusto integrado por un FPGA Xilinx, que se combina con un microprocesador de última generación Cortex-A9 (MPU), de alto rendimiento y bajo consumo de energía. Tiene una RAM DDR3 de alta velocidad, capacidades para video y audio e interfaz Ethernet. Contiene todas las herramientas necesarias para usar la tarjeta con una computadora que ejecute Windows. En la Figura 2 se muestra la tarjeta Zync-7000 SoC utilizada en el desarrollo del sistema y en la Figura 3 se muestra el diagrama de bloques del sistema de procesamiento de tarjeta.

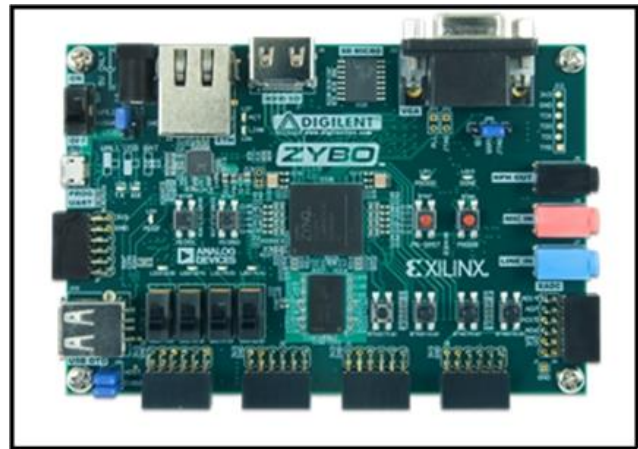


Figura 2. Tarjeta Zync-7000 SoC

B. Módulo de entrada de video

Para la entrada de video y captura de imágenes, tanto para generar las plantillas como para el reconocimiento de caracteres, se utilizó una cámara de video, como la mostrada en la Figura 4, conectada al puerto mini USB de la tarjeta Zync-7000 SoC. Las imágenes capturadas se convirtieron a formato de mapa de bits (bmp) y se

almacenaron en la tarjeta Micro SD que forma parte de los recursos del microprocesador.

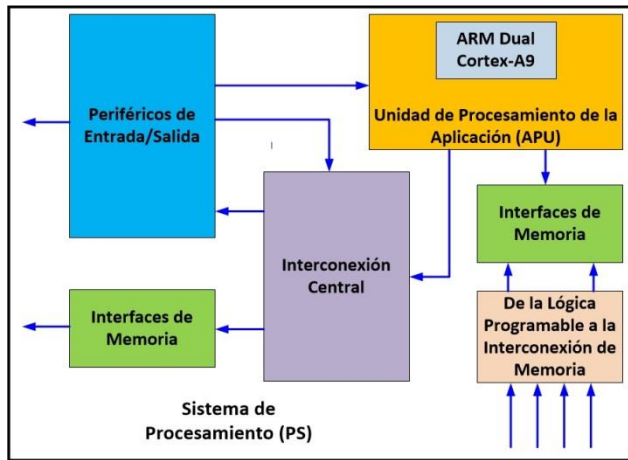


Figura 3. Bloques funcionales de la Tarjeta Zynq-7000 SoC



Figura 4. Cámara de video con conector mini USB

La estructura del archivo bmp generado consta de cuatro elementos: el encabezado del archivo, el encabezado de información del mapa de bits (también llamado encabezado de información), la paleta de la imagen y el cuerpo de la imagen. Se confirmó que las imágenes sean consistentes en calidad y apariencia. Se capturaron imágenes de letras y dígitos decimales a varias distancias para asegurar que sean distinguibles en la imagen, lo cual se explica en las pruebas y resultados obtenidos.

C. Módulo de entrada de audio

Para la entrada de audio se utilizó un micrófono, como el mostrado en la Figura 5, conectado a un puerto USB de la tarjeta Zynq-7000 SoC. Esto sirvió para grabar y registrar en la memoria SD el fonema de cada letra y dígito a reconocer.

D. Módulo de salida de audio

Para la salida de audio se utilizó un altavoz, como el mostrado en la Figura 6, conectado a otro puerto USB de

la tarjeta Zynq-7000 SoC. Por este medio se reproducen los fonemas previamente grabados y registrados al reconocer una letra o dígito.



Figura 5. Micrófono con conector USB



Figura 6. Altavoz con conector USB

E. Módulo de reconocimiento

Las plantillas generadas se usan para probar la coincidencia de la imagen capturada con una plantilla. Esto se hace utilizando las funciones existentes en la biblioteca de funciones de código abierto OpenCV, instalada y configurada en la tarjeta Zynq-7000 SoC. Particularmente, se utilizó la función *matchTemplate*, con la cual se genera una matriz de coeficientes de correlación para una imagen, una plantilla y un método de coincidencia. Estos coeficientes de correlación se analizaron para determinar el valor máximo del coeficiente de correlación y la ubicación de cada letra o dígito. La comparación de estos coeficientes máximos entre letras o dígitos permitió determinar la mejor letra o dígito coincidente.

En (1) se indica la función matemática usada para calcular el coeficiente de correlación, R , en cada ubicación (x, y) usando la plantilla T en la imagen I , la cual fue implantada utilizando OpenCV.

$$R(x, y) = \frac{\sum_{x', y'} (T'(x', y') \cdot I'(x + x', y + y'))}{\sqrt{\sum_{x', y'} T'(x', y')^2 \cdot \sum_{x', y'} I'(x + x', y + y')^2}} \quad (1)$$

I. RESULTADOS

Para probar la funcionalidad del sistema se llevaron cabo dos grupos de pruebas. El primer grupo tuvo como objetivo determinar el mejor tamaño de letras y dígitos a usar para generar las plantillas del método de comparación y llevar cabo el reconocimiento. Para realizar estas pruebas se seleccionó el tipo de letra Arial negra y se imprimieron 8 conjuntos de 37 caracteres cada uno. Cada conjunto estuvo compuesto de las 27 letras mayúsculas del alfabeto español, incluyendo la Ñ, y los 10 dígitos decimales. El tamaño de los caracteres de cada conjunto fue: 50, 100, 150, 200, 250, 300, 350 y 400 puntos, respectivamente. A continuación, se capturó, a través de la cámara de video, la imagen de cada caracter de los 8 conjuntos a una distancia de 60 centímetros. Las pruebas de este grupo consistieron en capturar 500 imágenes de caracteres y llevar a cabo el reconocimiento, tomando en cuenta que algunos caracteres a reconocer se repitieron. El reconocimiento inició usando primero el conjunto de plantillas de 50 puntos, posteriormente usando el conjunto de plantillas de 100 puntos, y así sucesivamente, hasta probar el reconocimiento utilizando cada conjunto de plantillas.

Los resultados mostraron, que, utilizando el conjunto de 50 puntos, de 500 caracteres capturados, se reconocieron 275, es decir, el 55%. Usando el conjunto de 100 puntos, de 500 caracteres capturados, se reconocieron 300, el 60%. De tal forma que se logró la mejor exactitud, 99.6%, al reconocer 498 caracteres de 500, cuando el tamaño del caracter de las plantillas fue mayor o igual a 300 puntos como se indica en la gráfica de la Figura 7.

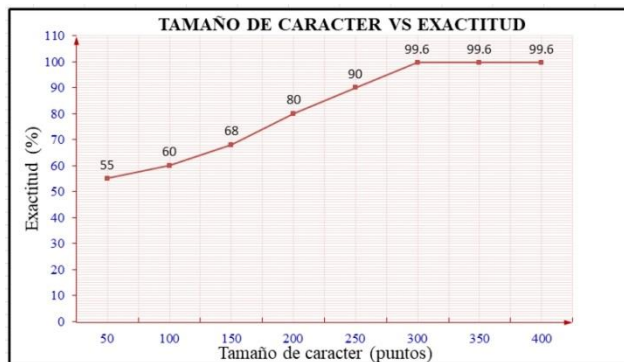


Figura 7. Exactitud lograda en el reconocimiento

El segundo grupo tuvo como objetivo determinar la mejor distancia de captura de la imagen de la plantilla y la

del caracter a reconocer para obtener el menor tiempo de respuesta del sistema. Las pruebas consistieron en realizar las capturas a diferentes distancias. Los resultados mostraron que a distancias entre 10 y 110 centímetros, el tiempo de reconocimiento fue 100 ms, el cual se incrementó hasta 600 ms al llevar a cabo las capturas a 110 centímetros, como se indica en la gráfica de la Figura 8. Para distancias mayores a esta última se consideró no continuar probando debido a que 600 ms es un tiempo alto de reconocimiento. Un problema que se presentó, a distancias mayores a 110 centímetros, fue que en algunas ocasiones el sistema no pudo distinguir entre la letra I y el número 1 y entre la letra O y el número 0.

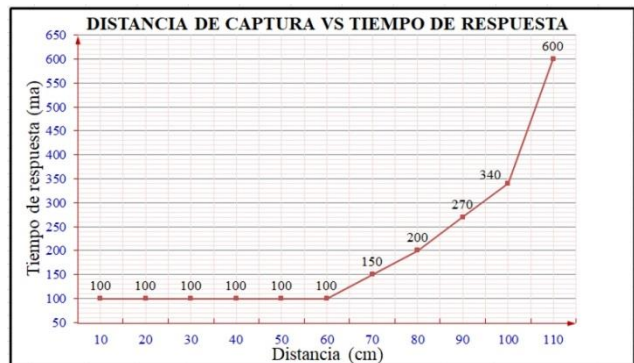


Figura 8. Distancia de captura y tiempo de respuesta

II. DISCUSIÓN, CONCLUSIÓN Y RECOMENDACIONES

La exactitud lograda en este trabajo fue 99.6%, mayor al de otras investigaciones realizadas, pero debe considerarse que los algoritmos y la aplicación llevada a cabo es esas investigaciones son más sofisticados, por ejemplo el método que usa el algoritmo del Tesseracto logró una exactitud de 88% y el que usa proyecciones horizontales y verticales logra 98.06%. También debe considerarse que existen algoritmos con mayor exactitud que la lograda en esta aplicación, como por ejemplo los que utilizan visión por computadora los cuales logran una exactitud de 99.99% cuyo uso no fue explorado en este trabajo. En cuanto al tiempo de respuesta, las investigaciones realizadas hasta el momento han demostrado que el menor tiempo, del orden de 10 a 11 ms, se logra implantando OCR a través de un FPGA. En este trabajo se logró un tiempo de respuesta de 100 ms., el cual se considera aceptable para la aplicación creada.

Finalmente, el trabajo a desarrollar a futuro es extender la implantación aquí realizada para concatenar caracteres y realizar reconocimiento de palabras. En este caso, tal vez sea recomendable usar otro método de OCR, por ejemplo, el de entrenamiento de un clasificador en cascada para obtener mayor exactitud utilizando herramientas de código abierto cada vez más disponibles y eficientes.

III. REFERENCIAS

- [1] Peymani, K. y Soryani, M. (2017). "From machine generated to handwritten character recognition; a deep learning approach", in *Proceedings 3rd International Conference on Pattern Recognition and Image Analysis (IPRIA)*, pp. 243-247.
- [2] Joshi, D. S. y Risodkar, Y. R. (2018). "Deep Learning Based Gujarati Handwritten Character Recognition", in *Proceedings International Conference On Advances in Communication and Computing Technology (ICACCT)*, pp. 563-566.
- [3] Thulasi, N. P., y Francis, S. (2017). "Intelligent tool for Malayalam cursive handwritten character recognition using artificial neural network and Hidden Markov Model", in *Proceedings International Conference on Inventive Computing and Informatics (ICICI)*, pp. 595-598.
- [4] Odate, R. y Goto, H. (2015). "Fast and accurate candidate reduction using the multiclass LDA for Japanese/Chinese character recognition", in *Proceedings IEEE International Conference on Image Processing (ICIP)*, pp. 951-955.
- [5] Prameela, N., Anjusha, P. y Karthik, R. (2017). "Off-line Telugu handwritten characters recognition using optical character recognition", in *Proceedings International conference of Electronics, Communication and Aerospace Technology (ICECA)*, Volume: 2, pp. 223-226.
- [6] Akouaydi, H., Abdelhedi, S. y Njah, S. (2017). "Decision trees based on perceptual codes for on-line Arabic character recognition", in *Proceedings 1st International Workshop on Arabic Script Analysis and Recognition (ASAR)*, pp. 153-157.
- [7] Pachpande, S. y Chaudhari, A. (2017). "Implementation of devanagiri character recognition system through pattern recognition techniques", in *Proceedings International Conference on Trends in Electronics and Informatics (ICEI)*, pp. 717-722.
- [8] Patil, A. V. y Dhanvijay, M. M. (2015). "Engraved character recognition using computer vision to recognize engine and chassis numbers: Computer vision technique to identify engraved numbers", in *Proceedings International Conference on Information Processing (ICIP)*, pp. 151-154.
- [9] Li, Q., An, W. y Zhou, A. (2016). "Recognition of Offline Handwritten Chinese Characters Using the Tesseract Open Source OCR Engine", in *Proceedings 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Volume: 02, pp. 452-456.
- [10] Mahto, M. K., Bhatia, K. y Sharma, R. K. (2015). "Combined horizontal and vertical projection feature extraction technique for Gurmukhi handwritten character recognition", in *Proceedings International Conference on Advances in Computer Engineering and Applications*, pp. 59-65.
- [11] Zho, H., Zhu, G. y Peng, Y. (2016). "A RMB optical character recognition system using FPGA", in *Proceedings IEEE International Conference on Signal and Image Processing (ICSIP)*, pp. 539-542.
- [12] Yin, F., Liu, P. y Huang, L. (2015). "Lexicon-driven recognition of one-stroke character strings in visual gesture", in *Proceedings 13th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 421-425.
- [13] Jing, Y., Youssefi, B. y Mirhassani, M. (2017). "An efficient FPGA implementation of Optical Character Recognition for License Plate Recognition", in *Proceedings IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 1 - 4.
- [14] Park, H., Lee, C. y Lee, H. (2017). "Work-in-progress: optimizing DCNN FPGA accelerator design for handwritten hangul character recognition", in *Proceedings International Conference on Compilers, Architectures and Synthesis For Embedded Systems (CASES)*, pp. 1-2.
- [15] Ivanov, S., Stankov, S. y Nenov, T. (2018). "FPGA Based Neural Networks for Characters Recognition", in *Proceedings 20th International Symposium on Electrical Apparatus and Technologies (SIELA)*, pp. 1-3.
- [16] Yata, K., Koutaki, G. y Uchimura, K. (2015). "Image matching by eigen template method for multi-class classification", in *Proceedings 21st Korea-Japan Joint Workshop on Frontiers of Computer Vision (FCV)*, pp. 1-4.
- [17] Zhang, Q., Wu, J. y Li, C. (2018). "Study of the Effects of Non-Square Resolutions of Bistatic Sar on Template Matching Performance", in *Proceedings IEEE International Geoscience and Remote Sensing Symposium*, pp. 557-560.

IV. BIOGRAFÍA



Vega-Luna José Ignacio. Ingeniería Electrónica, UAM-Azcapotzalco, Cd. de México, 1985. Maestría en Ciencias de la Computación, UAM-Azcapotzalco, Cd. de México, 1990. Labora actualmente en el área de Sistemas Digitales del Departamento de electrónica de la UAM-Azcapotzalco. Sus líneas de trabajo son: aplicaciones de microprocesadores y microcontroladores y sistemas operativos. El M. en C. Vega realiza investigación con redes inalámbricas de sensores y actuadores.



Salgado-Guzmán Gerardo. Ingeniería Electrónica, UAM-Azcapotzalco, Cd. de México, 1992. Labora actualmente en el Departamento de Electrónica de la UAM-Azcapotzalco. Sus líneas de trabajo son: aplicaciones de microprocesadores y microcontroladores y sistemas operativos. El Ing. Salgado realiza investigación con redes inalámbricas de sensores y actuadores.



Lagos-Acosta Mario Alberto. Ingeniería Electrónica, UAM-Azcapotzalco, Cd. de México, 1992. Labora actualmente en el Departamento de Electrónica de la UAM-Azcapotzalco. Sus líneas de trabajo son: aplicaciones de microprocesadores y microcontroladores y redes de computadoras. El Ing. Lagos realiza investigación con redes de computadoras y seguridad informática.



Cosme-Aceves José Francisco. Ingeniería Electrónica, UAM-Azcapotzalco, Cd. de México, 1985. Labora actualmente en el Departamento de Electrónica de la UAM-Azcapotzalco. Su línea de trabajo es lenguajes de descripción de hardware. El Ing. Cosme realiza investigación con sistemas embebidos y seguridad en redes de computadoras.



Sánchez-Rangel Francisco Rangel. Ingeniería Electrónica, UAM-Azcapotzalco, Cd. de México, 1987. Maestría en Ciencias de la Computación, UAM-Azcapotzalco, Cd. de México, 1999. Labora actualmente en el Departamento de Electrónica de la UAM-Azcapotzalco. Sus líneas de trabajo son: aplicaciones de microprocesadores y microcontroladores y lenguajes de descripción de hardware. El M. en C. Sánchez realiza investigación con redes de computadoras y sistemas embebidos.