

Herramienta para diseño automático de arquitecturas a medida basadas en redes neuronales para reconocimiento de patrones visuales

Lucas Leiva, Nelson Acosta, Martín Vázquez
 {lleiva, nacosta, mvazquez}@exa.unicen.edu.ar
 Facultad de Ciencias Exactas, INCA/INTIA, UNCPBA
 Tandil, 7000, Argentina

ABSTRACT

El trabajo propone la construcción de una herramienta con interfaz intuitiva capaz de generar arquitecturas hardware de reconocimiento de patrones a partir de un conjunto de especificaciones de alto nivel ingresadas por el usuario. La salida de la herramienta es código de descripción hardware sintetizable (VHDL) que se utiliza para la configuración de una FPGA. Se propone que la herramienta facilite todas las etapas del desarrollo de sistemas de reconocimiento de patrones visuales.

Keywords: VHDL, FPGA, Reconocimiento de Patrones, Redes Neuronales, Sistemas Embebidos.

1. INTRODUCCIÓN

Los sistemas de reconocimiento de patrones abarcan un amplio rango de aplicaciones, desde sistemas de control en procesos industriales (inspección de control de calidad[1], control de obstrucción en líneas de producción[2], clasificación[3]) hasta sistemas de seguridad [4][5].

Un patrón es una entidad a la que se le puede dar un nombre y que está representada por un conjunto de propiedades medidas y las relaciones entre ellas (vector de características) [6]. Por ejemplo, un patrón podría ser una señal sonora en donde su vector de características es el conjunto de coeficientes espectrales extraídos a partir de la misma (espectrograma). Una característica es una cualidad crítica dentro de un patrón, dado que el reconocimiento de patrones es llevado a cabo mediante su medición y comparación.

Un sistema de reconocimiento de patrones está compuesto por tres capas principales: adquisición de datos, extracción de características y clasificación (Fig. 1)

En el caso de reconocimiento de patrones en imágenes la primera de las etapas generalmente es llevada a cabo mediante un dispositivo de captura de imagen, que se encarga de transformar información obtenida del mundo real en un vector numérico que contiene los valores muestreados y cuantificados.

La etapa de selección y extracción de características es de suma importancia en un sistema de reconocimiento de patrones. Requiere un profundo análisis de los patrones para determinar cuales mediciones son las cruciales en la identificación de

las diferentes categorías bajo investigación. Durante esta etapa se aborda la recolección de información relevante proveniente de los dispositivos sensores para el proceso de clasificación.

El objetivo final de un sistema de reconocimiento de patrones es la asignación automática de una categoría (o clase informacional) a cada uno de los patrones de entrada. El proceso de construcción de la etapa de clasificación suele denominarse como aprendizaje o entrenamiento, pudiendo ser este *supervisado*, en donde se realiza a partir de un conjunto de patrones del que no se conoce su clase cierta; o *no supervisado*, los cuales requieren de la disposición de un conjunto de patrones, denominado conjunto de entrenamiento, de los cuales se conoce su clase cierta.

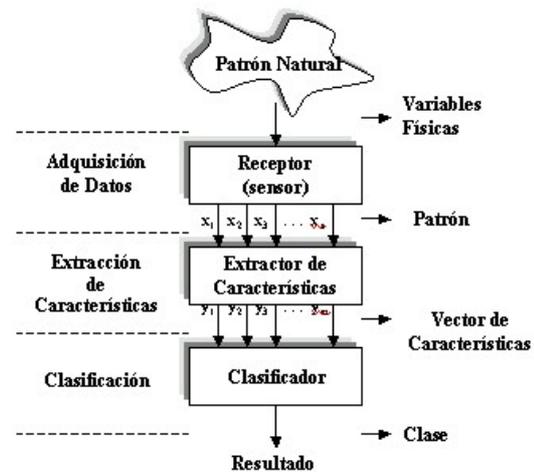


Fig. 1: Etapas de un sistema de Reconocimiento de Patrones

Este esquema (Fig. 1) no debe verse como los pasos a seguir en la construcción de un sistema de reconocimiento de patrones, sino más bien desde un punto de vista funcional: la entrada es un patrón natural y el resultado es una etiqueta. Además, no debe entenderse que todos los sistemas de reconocimiento de patrones deben incorporar todas estas unidades, ni siquiera que éstas deban estar tan claramente separadas.

De todos los clasificadores estudiados se ha decidido utilizar redes neuronales, fundamentalmente por su simplicidad y robustez. Así, para la clasificación de

patrones se encuentran las redes RBF¹, Feedforward, Hopfield, Perceptrón, Vector Quantization entre otras[7]. La clasificación está mecanizada mediante un neuro-clasificador, el cual es entrenado utilizando diferentes clases de datos como entradas, junto con sus categorías. Una red clasificadora mapea un vector de entrada a una clase representada, produciendo como señal de salida la pertenencia de la entrada a la misma. Además la red puede indicar el grado de aceptación de la entrada a la categoría, por lo cual no se limita a una señal binaria.

La implementación de técnicas neuronales en hardware dedicado (neurochips) posee la capacidad de proveer respuesta y entrenamiento en tiempo real para redes con grandes números de neuronas y sinapsis. Además brindan soluciones físicamente robustas para aplicaciones en donde no es posible instalar una PC. Existen en el mercado varios neurochips, tanto desarrollados por institutos de investigación como comerciales. Dentro de los comerciales se encuentran los procesadores ZISC78[8] desarrollados por IBM, los cuales son fácilmente escalables y cuentan con 78 neuronas implementando redes RBF.

Se propone la construcción de una herramienta que asista en todas las etapas de la construcción de sistemas de reconocimiento de patrones visuales hardware mediante el uso de redes neuronales, más precisamente procesadores ZISC78. La salida de la herramienta es código VHDL y cuenta con la lógica necesaria para llevar a cabo el reconocimiento. Este código es utilizado por el usuario para la configuración de una FPGA, que es la encargada de establecer la coordinación entre los diferentes dispositivos (red neuronal, cámara y bancos de almacenamiento). Las FPGAs son dispositivos programables, con una potencia similar a los ASICs a un menor costo, por lo cual se ajustan adecuadamente a este tipo de sistemas.

2. HERRAMIENTA PROPUESTA

La herramienta propuesta permite definir proyectos en donde cada uno de ellos cuenta con una única FPGA. Se ha decidido trabajar con una Spartan II-E de Xilinx con 200K puertas como plataforma de prueba. Sin embargo, el objetivo es independizar la generación de los recursos disponibles, de tal manera de proveer la suficiente flexibilidad como para soportar nuevas familias y nuevos fabricantes. Por otra parte el código generado es independiente del sintetizador ya que éste cumple con el estándar VHDL[9].

Dentro de un proyecto (Fig. 2), la FPGA tiene conexión con un dispositivo de captura de imagen, con cierto número procesadores ZISC y bancos de almacenamiento, definidos por el usuario, para los cuales la herramienta materializa un controlador.

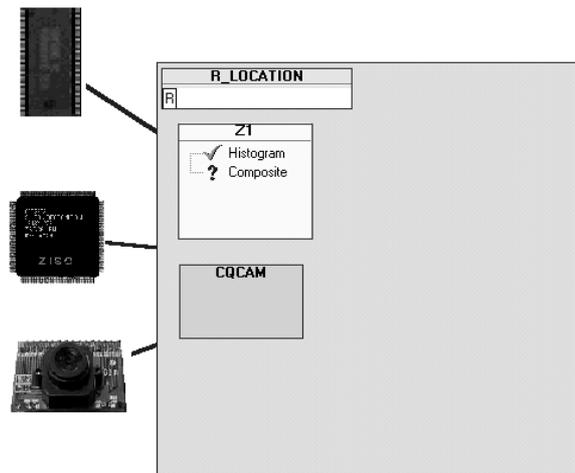


Fig. 2: Ejemplo de proyecto utilizando un dispositivo de captura, un banco de memoria y un ZISC78.

Durante la especificación de la fase de adquisición de datos, el usuario indica cuáles son las componentes deseadas del espacio de representación y los dispositivos en los cuales se almacenarán dichas componentes. Es decir, si la aplicación final cuenta con contrastes de rojos, no es necesario el almacenamiento de las componentes azul ni verde.

Con respecto a la especificación de la etapa de extracción de características, los controladores de los dispositivos ZISC poseen extractores de características (proyecciones), que pueden ser utilizados en el reconocimiento de varias características simultáneamente (ej: color, forma y textura). En el caso de que se cuente con un solo ZISC entrenado con dos proyecciones, el mismo trabajará de manera serial mediante la utilización de un contexto diferente para cada una de ellas. Por otro lado; si se cuenta con dos ZISC, cada uno entrenado con una proyección diferente, éstos trabajarán en paralelo [10].

La herramienta define un lenguaje de alto nivel del tipo vectorial y su compilador capaz de producir evaluación de imágenes tanto en software como en hardware permitiendo, de esta manera, la definición de las proyecciones.

Se cuenta con un entrenador en donde cada una de las proyecciones es entrenada definiendo las categorías que la conforman. Este entrenamiento puede ser automático, a partir de una categoría y un conjunto de subimágenes que la conforman, o asistido por el usuario, en donde a partir de una imagen de entrada se indica la categoría de una cierta región. Además, es posible la evaluación de una imagen, de tal manera de conocer si se ha alcanzado el entrenamiento deseado.

Para cada uno de los dispositivos se cuenta con una plantilla de información, en donde se detallan datos específicos acerca del dispositivo (ej: la capacidad de un banco de memoria específico, la resolución y formato de captura de una cámara), y la descripción del controlador en VHDL, soportando la

¹ RBF: Radial Basis Function

instanciación de ciertos rótulos durante la generación de código (ej: estado de las neuronas de un procesador ZISC78). Estas plantillas brindan un alto grado de flexibilidad en cuanto al conjunto de dispositivos soportados y al código generado.

Para permitir una mayor independencia del hardware, se incluye la posibilidad de utilizar bancos de neuronas RBF generados automáticamente por la herramienta que se encontrarán embebidos en la FPGA. Esto evita la necesidad de contar con dispositivos ZISC78 en la implementación de una aplicación. Estos bancos de neuronas pueden operar secuencialmente o en paralelo, según las necesidades del usuario. Si bien una red neuronal operando en modo paralelo posee una mayor performance, los recursos lógicos utilizados son mayores. De esta manera existe una limitación con respecto al número de neuronas embebidas que depende directamente de las capacidades de la FPGA.

Aprovechando los recursos existentes con los que cuentan las FPGAs actuales, la herramienta brinda la posibilidad de embeber los bancos de almacenamiento en los bloques de memoria dedicada con los que cuentan estos dispositivos. La instanciación de estos bloques depende de la familia y el fabricante, por esto se utilizan archivos que contienen este tipo de información para cada dispositivo programable.

Estas últimas características brindan la posibilidad de crear un sistema embebido, más precisamente un sistema *on-a-chip* (Fig. 3), permitiendo montar una aplicación con sólo contar con una cámara y una FPGA.

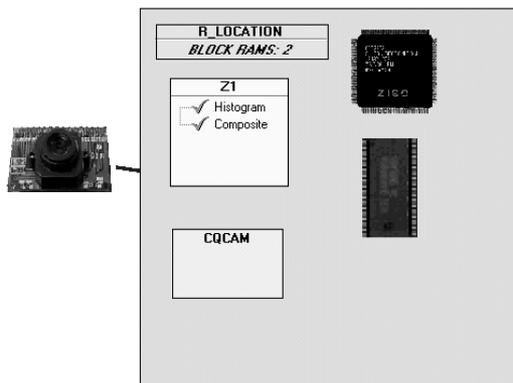


Fig. 3: Sistema *on-a-chip*.

Con respecto a la salida del sistema, ésta indica la posición dentro de la imagen y super-categoría a la que pertenece. Esta super-categoría es definida por el usuario a partir de las combinaciones que establece de los resultados de las proyecciones. Por ejemplo, considerando dos proyecciones en donde una identifica la forma del patrón y la otra su color, si el sistema es de clasificación de frutas, el usuario puede definir que las categorías “redondo” y “naranja” define a la super-categoría “naranja”, y “ovalado” y “marrón” define a “kiwi”.

Finalmente, la herramienta provee la asignación de pines de entrada/salida que conecta los dispositivos con su controlador correspondiente. Los pines varían de acuerdo al empaquetado de la FPGA y del modelo. Esto obliga a que esta información se almacene en los archivos de especificación del dispositivo. En el caso que la plataforma de implementación posea conexiones establecidas, el usuario deberá conocer cómo están mapeadas dichas conexiones.

5. ARQUITECTURA HARDWARE

La arquitectura generada se compone de (Fig. 4): i) controladores de los dispositivos utilizados, ii) componente de exploración de imagen, iii) extractores de características definidos (proyecciones), iv) módulo de resultados y v) unidad de control.

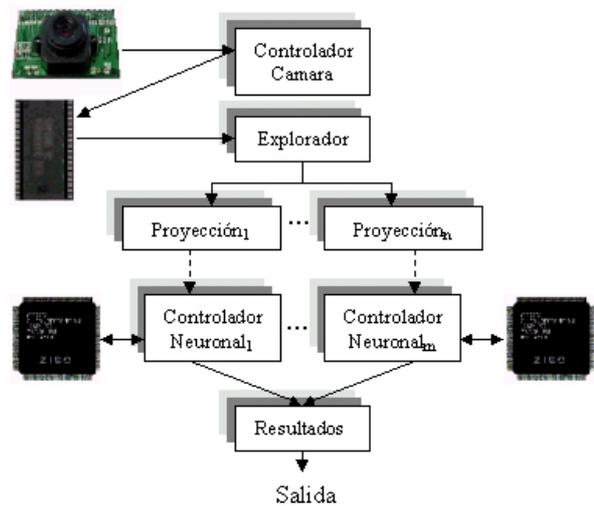


Fig. 4: Diagrama comportamental de la arquitectura

Operacionalmente los controladores poseen dos fases, la primera de ellas es una fase de configuración del dispositivo físico que se lleva a cabo durante el inicio del sistema. La segunda es la fase funcional del controlador.

El componente de exploración realiza un barrido de la imagen, obteniendo subimágenes (regiones de interés) a partir de las cuales se extraen las características. Actualmente la herramienta cuenta con un único modo de exploración (*raster*), que realiza un barrido horizontal de la imagen con desplazamientos ajustables por el usuario en la herramienta.

Los extractores de características operan en paralelo y sus resultados son transmitidos a los controladores de las redes neuronales a los cuales pertenecen.

El componente de resultado combina los resultados de las clasificaciones, tal como lo dispuso el usuario, brindando la salida final del sistema.

La coordinación de las operaciones se lleva a cabo en la unidad de control.

4. ESTADO DEL PROYECTO

La herramienta actualmente se encuentra en las fases finales de generación de código intermedio. Se están testeando y optimizando los componentes generados automáticamente.

5. TRABAJO FUTURO

Teniendo en cuenta las prestaciones de flexibilidad con las que cuenta la herramienta, se pretende ampliar su catálogo de dispositivos, ya sean cámaras, unidades de almacenamiento, neurochips o dispositivos programables.

Por otro lado, se propone la inclusión de información estadística durante el proceso de aprendizaje, de tal manera que la herramienta pueda ser utilizada en la formación de expertos en clasificación de patrones.

Además se propone incorporar nuevos tipos de exploraciones de la imagen, también parametrizables.

Finalmente, el mayor objetivo es incluir dentro de la FPGA entrenamiento no supervisado. Mediante el uso del desvío estándar de la distancia se puede definir un cierto límite donde, si un vector de características lo excede, el vector clasificado debe ser aprendido por la red neuronal artificial. Esta característica es de suma importancia en sistemas de seguimiento de objetivos.

6. REFERENCIAS

- [1] General Vision, Beer fill level inspected with ZiCAM Industry, Application Note, www.general-vision.com, 2005.
- [2] General Vision, Pisces VMK delivers its fileting machines with a smart camera controlling proper fish feeding and preventing jamming, Application Note, www.general-vision.com, 2005.
- [3] IVP, Sorting potatoes with an artificial eye, Case Study 1, www.ibp.se, 1998.
- [4] Strategic Page, Electronic Battlefield: Digital Lookouts, www.strategicpage.com, 10 de septiembre de 2005.
- [5] N.Malasne, F.Yang, M.Paindavoine, Real Time Implemetation Of A Face Tracking, Laboratory LE2I, University of Burgundy, 2003
- [6] S.Watanabe, Pattern Recognition: Human and Mechanical, Wiley, 1985.
- [7] Wolfram Research, Inc, Neural Networks, Wolfram Research, Inc, 2005
- [8] Silicon Recognition, ZISC: Zero Instruction Set Computer, Version 4.2, Silicon Recognition, Inc., 2002.
- [9] IEEE Std 1076, IEEE Standard VHDL Language Reference Manual, IEEE, 2000
- [10] N. Acosta, M. Vázquez, C. Aciti, L. Leiva, A Parallel Architecture for Neural-Network Processors, WMSCI 2005, U.S.A., 2005