

Toward pampas deer images classification using BCNN implemented on an FPGA

M. Vernengo ¹, J.M. Santos ¹ and J.I. Giribet ²

¹ UNAHUR, Argentina, ² UdeSA & CONICET, Argentina

Abstract. *This paper introduces a line of research for the classification of images of the Pampas deer (*Ozotoceros bezoarticus*), a threatened species according to the International Union for Conservation of Nature (IUCN). The classification is performed using a Binarized Convolutional Neural Network (BCNN) implemented on a platform that combines a CPU and an FPGA (Field Programmable Gate Array) within a single SoC (System on Chip), such as the AMD/Xilinx Zynq-7000, to accelerate computation through parallel processing. This research is part of a doctoral thesis focused on the development of embedded systems for wildlife monitoring, with the ultimate goal of achieving real-time detection on board Unmanned Aerial Vehicles (UAVs). By leveraging the parallel processing and low-power capabilities of the FPGA, we provide a high-performance solution for autonomous monitoring in remote habitats.*

Keywords: *Pampas deer (*Ozotoceros bezoarticus*), Binarized Convolutional Neural Networks (BCNN), Image classification using FPGA platforms.*

Hacia la clasificación de imágenes del venado de las pampas mediante BCNN implementadas en FPGA

M. Vernengo ¹, J.M. Santos ¹ y J.I. Giribet ²

¹ UNAHUR, Argentina, ² UdeSA y CONICET, Argentina

Resumen. Este trabajo presenta una línea de investigación para clasificar imágenes del venado de las pampas (*Ozotoceros bezoarticus*), especie amenazada según la Unión Internacional para la Conservación de la Naturaleza (UICN). La clasificación se realiza mediante una Red Neuronal Convolutiva Binarizada (BCNN) implementada en una plataforma que combina en un mismo SoC (*System on Chip*) una CPU y una FPGA (*Field Programmable Gate Array*), tal como el Zynq-7000 de AMD/Xilinx, con el fin de acelerar el cómputo mediante procesamiento paralelo. Esta investigación integra una tesis doctoral centrada en el desarrollo de sistemas embebidos para el monitoreo de fauna silvestre. El objetivo final es lograr la detección en tiempo real desde vehículos aéreos no tripulados (UAV). Al aprovechar el procesamiento paralelo y el bajo consumo del SoC CPU-FPGA, se obtiene una solución de alto rendimiento para el monitoreo autónomo en hábitats remotos.

Palabras clave: Venado de las pampas (*Ozotoceros bezoarticus*), Redes Neuronales Convolutivas Binarizadas BCNN, Clasificación de imágenes con plataformas FPGA.

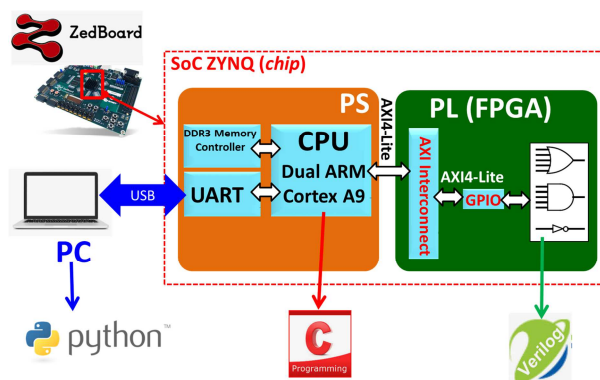
1 Introducción

La línea de investigación presentada en este trabajo busca desarrollar herramientas de *software* y *hardware* para la conservación del venado de las pampas (*Ozotoceros bezoarticus*), especie amenazada (González et al., 2016) con mortalidad vinculada a factores ambientales (Orozco et al., 2013).

El monitoreo en tiempo real mediante UAVs es eficaz, pero el procesamiento en plataformas convencionales sufre restricciones energéticas crónicas; a esto se suma que los ejemplares ocupan pocos píxeles y sufren oclusión o mala iluminación. Aunque trabajos como los de Lekhadia et al. (2025) y Adhikari et al. (2025) abordan estos desafíos, ninguno contempla las restricciones de imágenes cenitales, la relación altura-desempeño ni la eficiencia energética, aspectos centrales de este trabajo.

Como solución, se proponen Redes Neuronales Convolucionales Binarizadas (BCNN) de 1 bit (Courbariaux et al., 2016; Rastegari et al., 2016). Si bien frameworks como FINN (Umuroglu et al., 2016) facilitan la síntesis de alto nivel (HLS) en SoC Zynq, investigaciones como la de Alam et al. (2022) reportan caídas de velocidad superiores al 30% frente a diseños a medida. Por ello, considerando que el chip XC7Z020 posee recursos limitados en comparación con la plataforma ZC706 de referencia, se propone una BCNN diseñada íntegramente en Verilog.

Figura 1. Esquemas de interconexión y programación de las plataformas utilizadas para implementar una neurona artificial y en trabajos futuros, también para los PEs.



2 Propuesta de arquitectura de *hardware*

El desarrollo de esta línea de investigación se asienta sobre la base del diseño y simulación de una arquitectura de *hardware* con una FPGA con el objetivo de demostrar la posibilidad de sintetizar hasta 1024 elementos de procesamiento (PE por sus siglas en inglés) para convolución binarizada sin agotar los recursos del *chip* XC7Z020 de la plataforma Zynq-7000. El objetivo es brindar un margen para implementar otras operaciones requeridas por las BCNN, como se describe más adelante en la sección 4 y que son objetivo futuro de esta línea de investigación.

La Fig. 1 muestra la prueba de concepto de la interconexión con la que se implementó una neurona individual ó “perceptrón” para clasificar imágenes simples (rectas horizontales o verticales), validando el envío de pesos y píxeles entre la PC (donde se realizó el entrenamiento) y la FPGA. La CPU del SoC actúa como interfaz vía bus AXI GPIO interno del *chip*. La transferencia actual en bloques de 16 bits se

reemplazará a a futuro por acceso directo a memoria (DMA por su sigla en inglés) para acelerar el procesamiento.

3 Materiales y métodos

La clasificación de imágenes con cérvidos es un proceso que podemos describir como dividido en 3 etapas y permite discriminar al cérvido frente a imágenes de fondos de vegetación sin especímenes de animales (de aquí en más “fondos”):

- i. **Conjunto de Datos** incluye 244 imágenes de UAV, todas con al menos un cérvido y de 3648 x 4864 píxeles cada una (ver Fig. 2(a)), contándose además con máscaras con la ubicación de los ciervos que mediante el algoritmo de componentes conectados de Open CV (método `cv2.connected`) permitió extraer parches de 128 x 128 píxeles con ciervos (ver Fig. 2(b)) por un lado y de fondos por otro, consolidando un conjunto base de 4.540 muestras de fondos sin ciervos. Los parches se submuestrearon a 32 x 32 píxeles. El desbalance se mitigó mediante un sobremuestreo asimétrico de 1:18 con aumento dinámico de parches de la clase minoritaria (ciervos) con rotaciones, traslaciones y cambios de brillo, contraste y saturación.
- ii. **Entrenamiento y Binarización:** la red se entrena en un entorno Python/TensorFlow mediante la librería LARQ. La arquitectura se basa en la topología CNV propuesta por Umuroglu et al. (2017) para CIFAR-10, inspirada en BinaryNet y VGG-16. Contiene tres bloques compuestos por dos capas consecutivas de convolución de 3x3 cada uno, seguidas de *batch normalization* y *maxpooling* de 2x2, operando con 64, 128 y 256 canales respectivamente. A esto le siguen dos capas completamente conectadas de 512 neuronas y una capa de salida *softmax* de 2 neuronas para la clasificación binaria entre cérvidos y fondos de vegetación. Se utilizó la función Focal Loss (Lin et al., 2017) con $\alpha = 0.90$ para penalizar falsos positivos, lo que permitió mitigar adicionalmente el desbalanceo de datos de las clases.
- iii. **Inferencia con SoC CPU/FPGA (aún no implementada):** los pesos e imágenes se transferirán secuencialmente hacia memorias distribuidas LUT (*Look Up Table*) de la FPGA (tal como previamente mencionado y validado para un perceptrón individual). La clasificación a bordo del UAV se realizará vía procesamiento paralelo con los PEs. La arquitectura optimiza área y consumo del *chip* reemplazando multiplicadores por sumadores y restadores, ya que los pesos son binarios (+1 y -1) y también lo son las activaciones (sólo la entrada de imágenes RGB se mantiene en 8 bits por canal).

4 Resultados de simulación y análisis de recursos

El diseño con Verilog fue verificado mediante simulación funcional tanto con Logisim para 4 PEs (ver Fig. 3) como con la herramienta Vivado para 1024 PEs, con coincidencia exacta con los cálculos manuales para mapas de características multicanal y con una precisión de 14 bits para asegurar un rango dinámico suficiente en la acumulación previa a la activación *reLU*. Debido a que la red opera con pesos

estrictamente binarios, las multiplicaciones aritméticas convencionales fueron reemplazadas por operaciones de suma y resta mediante selección de signo por lo que sólo uno de los 220 bloques DSP disponibles fue requerido (0,45%). La implementación física completa de los 1024 PEs en el SoC XC7Z020 consumió el 65,56% de las LUTs y el 21,96% de los Flip-Flops. Los buffers locales se mapearon de forma íntegra como LUTRAM, reduciendo a un 0% la utilización de BRAM.

Asimismo, el sistema cumplió con todas las restricciones de tiempo, alcanzando un cierre de diseño exitoso con un Worst Negative Slack (WNS) de 5,051 ns. Este grado de uso y margen temporal confirman la capacidad remanente del silicio para incorporar funciones como *MaxPooling* y *Batch Normalization*, esta última vía umbralización, evitando así operaciones de coma flotante.

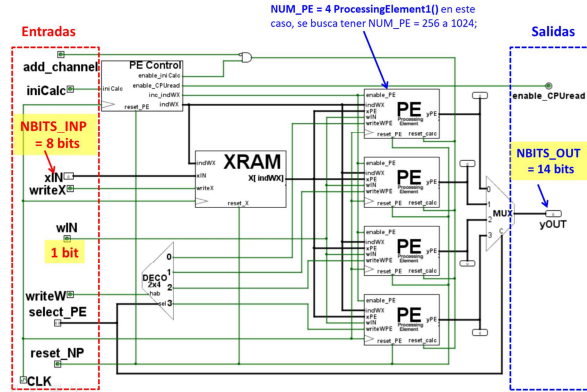
Al destinar un 20% de los parches al conjunto de validación ciega (908 fondos y 49 cérvidos), la BCNN con un umbral de 0,50 consolidó una exactitud del 99,58%. En la clase minoritaria, el modelo alcanzó una sensibilidad (*Recall*) del 100%, al identificarse los 49 ejemplares sin falsos negativos y una Precisión de 92,45% con solo 4 falsos positivos, lo que implica un *F1-Score* del 96,23%. El resultado demuestra alta capacidad de generalización ante datos independientes.

Figura 2. Ejemplo de imágenes del venado de la pampas utilizadas:

- Imagen (UAV) de 4864×3648 pixeles; flecha y recuadro agregados señalan el cérvido.
- Recorte de 128×128 pixeles de (a) submuestreado a 32×32 pixeles.



Figura 3. Esquema de 4 PE's simulado con Logisim. Los pixeles xIN de 8 bits son para los 3 canales RGB de entrada, pero se usa un sólo 1 bit en capas internas. Los pesos wIN son siempre de 1 bit (con el 0 representando -1).



5 Conclusiones

Esta línea de investigación busca consolidar una plataforma de *hardware* para realizar clasificaciones de imágenes con animales silvestres en entornos naturales que sirvan como herramienta para implementar algoritmos de localización e identificación de una especie en estudios realizados con UAV autónomos con eficiencia energética.

En la actual etapa de desarrollo del proyecto se ha logrado integrar exitosamente el flujo de datos PC-SoC y sintetizar y simular con la herramienta de desarrollo del fabricante una arquitectura de PEs de convolución binarizada bajo el paradigma BCNN. También se logró implementar el entrenamiento con BCNN para clasificación de imágenes del venado de las pampas, apto para permitir la transferencia de pesos sinápticos binarizados a la plataforma SoC CPU + FPGA.

Los próximos pasos consisten implementar las etapas adicionales de *MaxPooling* y *Batch Normalization* a la de convolución ya simulada y luego implementar la interconexión con la PC de entrenamiento y CPU+FPGA del SoC Zynq-7000 utilizando DMA, finalizando así la implementación completa de *hardware* y *software*.

Referencias

- Adhikari, B., Li, J., Michel, E. S., Dykes, J., Tseng, T. M. P., Tagert, M. L., & Chen, D. (2025). A comprehensive evaluation of YOLO-based deer detection performance on edge devices. *Electronics*. Obtenido de <https://doi.org/10.48550/arXiv.2509.20318>.
- Alam, M. A., Gregg, D., Gambardella, G., Blott, M. y Preusser, T. B. (2022). On the RTL Implementation of FINN Matrix Vector Compute Unit. En 2022 IEEE 33rd International Conference on Application-specific Systems, Architectures and Processors (ASAP) (pp. 147-154). IEEE. <https://doi.org/10.1109/ASAP54719.2022.00032>.
- Bannink, T., Bakhtiari, A., Hillier, A., Geiger, L., Tim, D. B., Overweel, L., Neeven, J. y Helwegen, K. (2020). LARQ Compute Engine: Design, benchmark, and deploy state-of-the-art binarized neural networks. arXiv. <https://arxiv.org/abs/2006.10170>.
- Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R. y Bengio, Y. (2016). Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1. arXiv. <https://arxiv.org/abs/1602.02830>.
- González, S., Jackson, J. E., III, & Merino, M. L. (2016). *Ozotoceros bezoarticus*. The IUCN Red List of Threatened Species 2016. <https://www.iucnredlist.org/species/15803/22160030>.
- Lekhadia, S., Ghoniya, P., Bala Vignesh Reddy, S., Panshala, A., Trivedi, D., & Singh, M. (2025). An edge computing approach for real-time wildlife detection and alert system using YOLOv8 on Raspberry Pi 5. *International Journal of Engineering Research & Technology (IJERT)*, 14(8). Obtenido de <https://doi.org/10.5281/zenodo.18103444>.
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2980–2988. <https://doi.org/10.1109/ICCV.2017.324>.
- Orozco, M. M., Marull, C., Jiménez, I. y Gürtler, R. E. (2013). Mortalidad invernal de ciervos de los pantanos (*Blastocerus dichotomus*) en humedales del noreste de Argentina. *Mastozoología Neotropical*, 20(1), 163-170.
- Rastegari, M., Ordonez, V., Redmon, J. y Farhadi, A. (2016). XNOR-Net: ImageNet classification using binary convolutional neural networks. arXiv. <https://arxiv.org/abs/1603.05279>.
- Sladevič, T. y Serackis, A. (2020). mNet2FPGA: A design flow for mapping a fixed-point CNN to Zynq SoC FPGA. *Electronics*, 9(11), 1823. <https://doi.org/10.3390/electronics9111823>.
- Umuroglu, Y., Fraser, N. J., Gambardella, G., Blott, M., Leong, P. H., Jahre, M. y Vissers, K. A. (2016). FINN: A framework for fast, scalable binarized neural network inference. En *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays* (pp. 1-10). ACM. <https://doi.org/10.1145/3020078.3021744>.