Accelerating Inference in Computer Vision Tasks with VCK190 and Vitis AI

Pedro-Pablo Gómez-González, Esteban Jove, and José Luis Calvo-Rolle

Research Group CTC, CITIC, Department of Industrial Engineering, Universidade da Coruña, 15403 Ferrol, A Coruña, Spain
Correspondence: pedro.pablo.gomez@udc.es

DOI: *https://doi.org/10.17979/spu.23.c22*

Abstract: The increasing use of Deep Neural Networks (DNNs) for real-time tasks has created a need for specialized hardware accelerators.This paper presents a study of the Xilinx VCK190 development board, a next generation Adaptive Compute Acceleration Platform (ACAP) for artificial intelligence tasks, representing a first approach to this technology at the Universidade da Coruña. The work details a complete workflow using Vitis AI, covering the quantization, compilation, and deployment of a neural network model. The results establish a solid technical baseline, intended to facilitate the integration of this advanced hardware into future academic and research projects.

1 Introduction

The rapid growth of Deep Neural Networks (DNNs) across various scientific fields, from image recognition to natural language processing, has led to a growing demand for specialized hardware platforms that can deliver high computational performance with energy efficiency, especially for real-time inference applications (C. Silvano et al., 2025). Field-Programmable Gate Arrays (FPGAs) have emerged as a viable alternative for accelerating inference in computationally intensive tasks due to their reconfigurable architecture, which allows for optimization of resource usage and minimization of latencies for specific neural networks (Li and Liewig, 2020).

In this context, the Xilinx VCK190 development board, based on the Versal Adaptive Compute Acceleration Platform (ACAP) architecture, represents a significant evolution beyond traditional FPGAs. The VCK190 integrates AI Engines, programmable logic, and ARM processors, enabling highly parallelized execution of artificial intelligence algorithms, including Deep Convolutional Neural Networks (CNNs). This heterogeneous architecture is specifically designed for workloads such as computer vision, edge computing, and real-time inference (AMD, 2024).

To bridge the gap between this advanced hardware and AI application development, Xilinx provides the Vitis AI development environment. This high-level toolkit abstracts the complexities of programmable logic, offering a streamlined workflow that includes tools for model quantization, compilation, and deployment. This allows developers, even those without deep hardware expertise, to implement accelerated AI solutions (AMD, 2023b).

The field of hardware acceleration for Deep Learning has been extensively surveyed, with works covering FPGA toolflows (Venieris et al., 2018), accelerators for the edge environment (Li and Liewig, 2020), and the broader landscape of HPC platforms including emerging technologies (C. Silvano et al., 2025). However, the VCK190 remains relatively unexplored in academic environments, partly due to its recent availability and cost. This paper addresses this void by presenting a complete and reproducible workflow, marking a first approach to this technology at the Universidade da Coruña. The main contributions of this work are:

1. An outline of the development environment setup for the VCK190 with Vitis AI 3.0 to ensure reproducibility.

2. The implementation of a versatile Python script for multi-architecture model quantization.

3. A performance evaluation of a deployed ResNet-18 model on computer vision tasks, analyzing both accuracy and inference speed (FPS) to establish a solid technical baseline for future projects.

The remainder of this paper is organized as follows: Section 2 details the methodology and the implemented workflow. Section 3 presents the conducted experiments along with their results. Finally, Section 4 provides the conclusions and outlines future work.

2 Methodology

This section details the hardware platform, software toolchain, and the end-to-end workflow implemented for quantizing and deploying a CNN on the target device. The methodology is presented as a case study focused on a ResNet-18 model, providing a reference for the practical application of the Vitis AI framework on the VCK190 platform.

2.1 Platform and Tools

The experiments were conducted on an AMD-Xilinx VCK190 evaluation board, which served as the target hardware. The board was configured using the pre-built PetaLinux 2022.2 image provided by the manufacturer [1]. This image is particularly suitable as it integrates the necessary drivers, the Vitis AI Runtime, and a high-performance DPUCVDX8G accelerator (C32B6CU1L2S2) by default, which utilizes a substantial amount of the device's programmable logic resources. The entire board setup process, which involves flashing the SD card and configuring the boot mode switches, was performed following the official Vitis AI Quick Start guide for the VCK190 AMD (2023a).

Model preparation was conducted on a host PC running Windows 11. To ensure compatibility with the toolchain, a Linux environment was configured using the Windows Subsystem for Linux (WSL) with an Ubuntu 20.04 distribution. The Vitis AI 3.0 toolchain was managed via Docker, utilizing the pre-built container for the PyTorch framework on a CPU architecture. To streamline the workflow, MobaXterm was used as the primary terminal client, centralizing access to both the local Ubuntu environment and remote connections to the target board (via UART and SSH). Furthermore, to compile the final C++ deployment application for the ARM processors on the target, the PetaLinux SDK cross-compiler was installed and configured on the host system. This is shown in Figure 1.

[1] The pre-built PetaLinux image for the VCK190 is available for download from the following link: *https://www.xilinx.com/member/forms/download/design-license-xef.html?filename=xilinx-vck190-dpu-v2022.2-v3.0.0.img.gz*

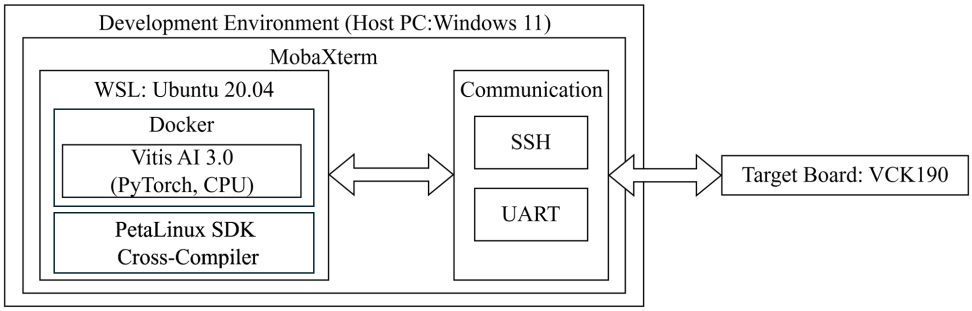

Figure 1: Architectural diagram of the development environment

2.2 Automated Quantization and Compilation Script

To automate the multi-stage and often iterative Vitis AI workflow, a versatile Python script, `model_quant.py`[2], was developed. This script serves as a modular and reusable framework that encapsulates the entire process, from loading a pre-trained model to generating the final deployable files. Its modular design supports a wide range of CNN architectures (Table 1) by dynamically adjusting input parameters, while its flexibility allows for rapid prototyping through control over batch size and dataset length. The script automates the main stages of the Vitis AI toolchain—hardware inspection, post-training quantization, and compilation—which are controlled via command line arguments. The complete operational logic is illustrated in the flowchart in Figure 2.

Table 1: CNN Architectures Supported by the Automation Script

Model Family	**Supported Architectures**
ResNet	`resnet18, resnet34, resnet50, resnet101, resnet152`
VGG	`vgg11, vgg13, vgg16, vgg19`
DenseNet	`densenet121, densenet161, densenet169, densenet201`
MobileNet	`mobilenet_v2, mobilenet_v3_large, mobilenet_v3_small`
EfficientNet	`efficientnet_b0, efficientnet_b1, efficientnet_b2`
SqueezeNet	`squeezenet1_0, squeezenet1_1`
Otras	`alexnet, inception_v3`

[2] The source code for the `model_quant.py` script is publicly available at: *https://github.com/pedropgg/Vitis-AI-Model-Quantization*

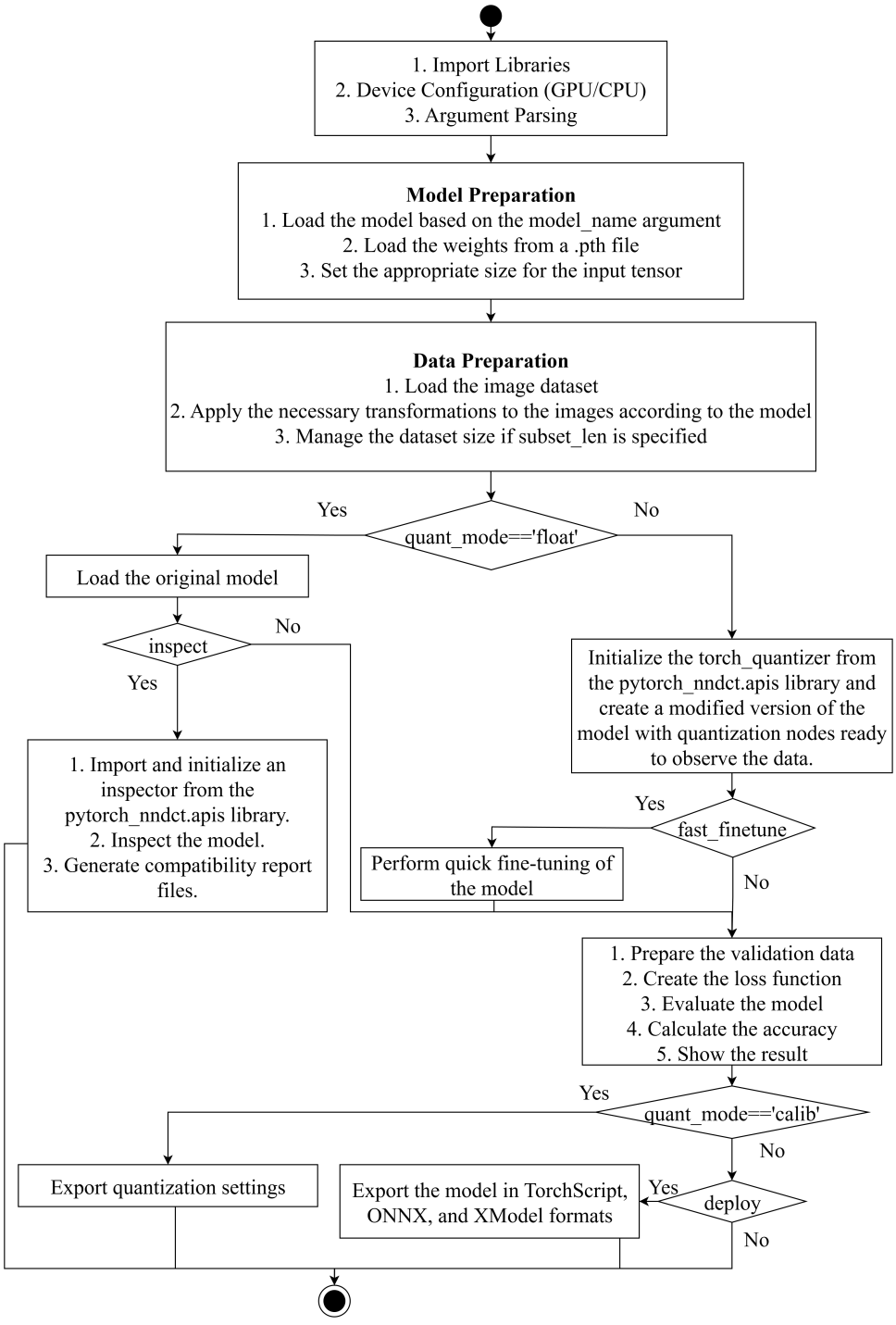

Figure 2: Flowchart of the `model_quant.py` script, illustrating the decision logic based on the operational mode.

2.3 Model Preparation Workflow

The model preparation workflow is a three-stage process (Quantization, Compilation, and Deployment), orchestrated by executing the `model_quant.py` script in different modes. A ResNet-18 architecture was selected as a standard benchmark for this study due to its balance of accuracy and complexity, enabling performance comparisons with other hardware acceleration studies.

1. **Quantization:** In the first stage, the user runs the script in `calib` mode. This employs a Post-Training Quantization (PTQ) approach to convert the 32-bit floating-point (FP32) model into a hardware-friendly 8-bit integer (INT8) representation. The script uses the `pytorch_nndct` API to perform calibration with a subset of the validation dataset, generating a configuration file with the optimal scaling factors.
2. **Compilation:** Next, the user executes the script in `test` mode with the `--deploy` flag. This triggers the Vitis AI Compiler, which processes the quantized INT8 model and compiles the network graph into a set of micro-coded instructions for the target DPUCVDX8G architecture. The output of this stage is a deployable `.xmodel` file.
3. **Deployment:** The final stage involves transferring the files generated on the previous stages from the host to the target VCK190 board. For on-board inference, a C++ application is compiled using the PetaLinux SDK cross-compiler. This application leverages the Vitis AI Runtime (VART) to load the transferred files, manage memory buffers, and schedule inference tasks on the DPU, using OpenCV for video I/O and visualization.

3 Experiments and Results

This section presents the results of the experimental evaluation of the implemented workflow using a ResNet-18 model. The evaluation was structured in two main phases:

- **Quantization Accuracy Evaluation:** A comparison of the model's predictive accuracy before and after PTQ, evaluated on the ImageNet validation dataset.
- **Device Performance Evaluation:** A throughput measurement in Frames Per Second (FPS) of the deployed model (INT8) on the VCK190 across different video processing scenarios.

3.1 Quantization Accuracy Evaluation

The results of the accuracy comparison between the FP32 and INT8 models are summarized in Table 2.

Table 2: Accuracy Comparison of FP32 vs. INT8 Model

Model Precision	**Top-1 Accuracy (%)**	**Top-5 Accuracy (%)**
FP32 (Original)	69.99	88.76
INT8 (Quantized)	69.54	88.78

The data shows a negligible accuracy loss of only 0.45% for the Top-1 metric, validating the effectiveness of the quantization process for this model.

3.2 Device Performance Evaluation

Pre-recorded Video File: To evaluate the system's performance using a video stored on the target device, two versions of the same pre-recorded video file were tested. When processing the version encoded at 60 FPS, the system achieved a sustained end-to-end performance of approximately 55 FPS (Figure 3). To further analyze the system's maximum throughput, the experiment was repeated with a version encoded at 240 FPS. Under this high-data-rate condition, the inference performance increased significantly, reaching approximately 90 FPS (Figure 4).

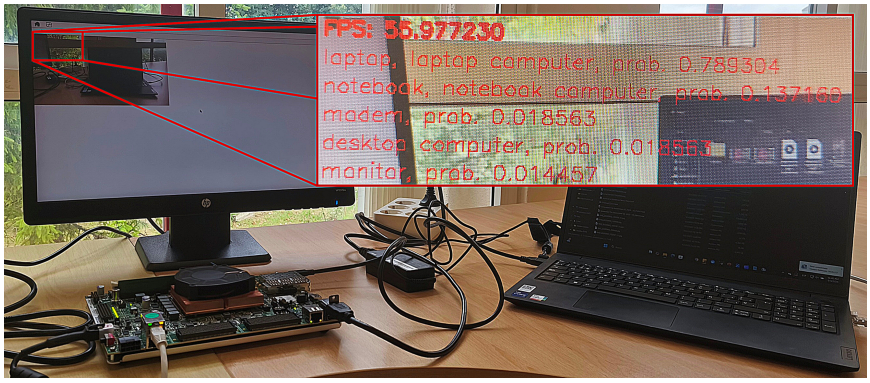

Figure 3: Inference on a 60 FPS video file. The monitor shows the classification ('laptop') and system performance (55 FPS).

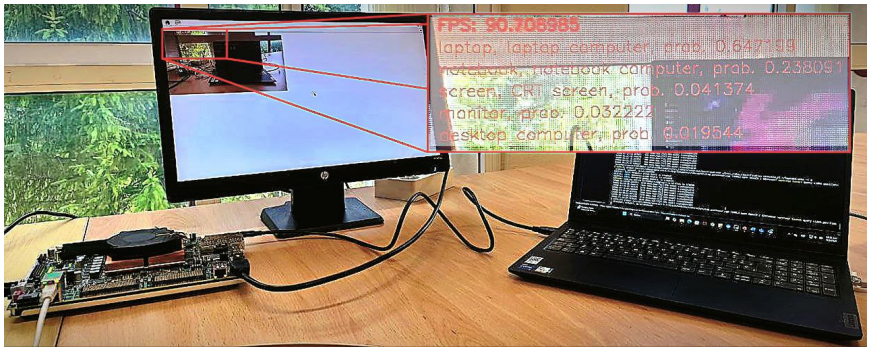

Figure 4: Inference on a 240 FPS video file. System throughput increases to 90 FPS.

Real-Time RGB Camera: Using the standard color (RGB) sensor of an Intel RealSense D455 camera [3], the system's performance was 15 FPS, a result limited by the camera's capture rate. The visual output of this test is shown in Figure 5.

[3] The technical specifications for the Intel RealSense D400 series cameras are available at: *https://www.intelrealsense.com/wp-content/uploads/2020/06/Intel-RealSense-D400-Series-Datasheet-June-2020.pdf*

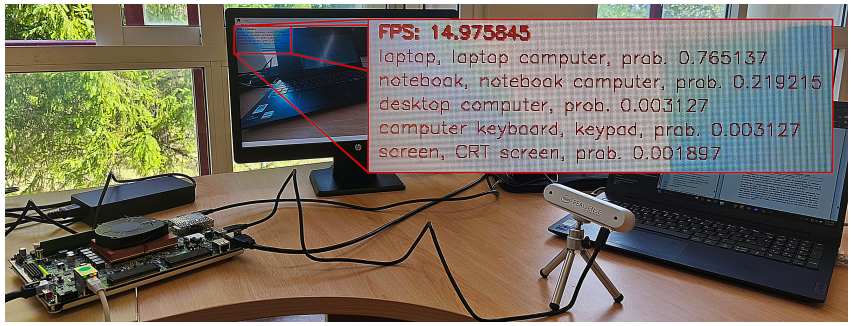

Figure 5: Real-time classification using the color (RGB) sensor of the Intel RealSense D455 camera. The camera limits performance to 15 FPS.

Real-Time Depth Camera: Using the depth sensor from the same camera, the performance increased to 30 FPS, again reflecting the sensor's specific frame rate for that mode. Figure 6 captures the output from this inference task.

Figure 6: Output from the RealSense D455 depth sensor. System performance increases to 30 FPS.

The results show that the inference speed in real-world scenarios is conditioned by the input source.

4 Conclusions and Future Work

This paper presented and validated a Vitis AI workflow for DNN acceleration in computer vision tasks on the VCK190 ACAP using a ResNet-18 case study. The results confirm the VCK190 is a highly effective platform for edge applications, achieving up to 90 FPS with a negligible 0.45% accuracy loss at a modest 21 W power consumption [4]. This efficiency (4.28 FPS/W) contrasts with server GPUs like the NVIDIA Titan X Pascal, which require non-real-time batching and greater power to achieve higher raw throughput [5]. A key finding is that the system's performance is ultimately limited by the I/O bandwidth of peripherals, not by the DPU's computational power.

The established framework and versatile Python script provide a solid foundation for future work, which will focus on applying this workflow to more complex models and exploring other optimization techniques on the ACAP architecture.

[4] The power consumption estimate was obtained using the Xilinx BEAM (Board Evaluation and Management Tool)

[5] For context, detailed ResNet-18 performance benchmarks on other hardware platforms, including the GPU mentioned, can be found at: *https://github.com/jcjohnson/cnn-benchmarks*

Acknowledgements

The CITIC, as an accredited center of excellence within the Galician University System and a member of the CIGUS Network, receives subsidies from the Ministry of Education, Science, Universities, and Vocational Training of the Xunta de Galicia. It is also co-financed by the EU through the Galicia 2021-2027 FEDER operational program (Ref. ED431G 2023/01).

Xunta de Galicia. Aid for the consolidation and structuring of competitive research units, GPC (ED431B 2023/49).

This activity is carried out in execution of the Strategic Project "Critical infrastructures cybersecure through intelligent modeling of attacks, vulnerabilities and increased security of their IoT devices for the water supply sector" (C061_/23), the result of a collaboration agreement signed between the National Institute of Cybersecurity (INCIBE) and the University of A Coruña. This initiative is carried out within the framework of the funds of the Recovery, Transformation and Resilience Plan, financed by the European Union (Next Generation), the project of the Government of Spain that outlines the roadmap for the modernization of the Spanish economy, the recovery of economic growth and job creation, for the solid, inclusive and resilient economic reconstruction after the COVID-19 crisis, and to respond to the challenges of the next decade.

Bibliography

AMD. Quick start example for vck190 — Vitis AI user guide 3.0. *https://xilinx.github.io/Vitis-AI/3.0/html/docs/quickstart/vck190.html*, 2023a. [Online; accessed 20-june-2025].

AMD. Vitis ai overview — Vitis AI user guide (ug1414). *https://docs.amd.com/r/3.0-English/ug1414-vitis-ai/Vitis-AI-Overview*, 2023b. [Online; accessed 5-June-2025].

AMD. Vck190 evaluation board user guide — AMD documentation. *https://docs.amd.com/r/en-US/ug1366-vck190-eval-bd*, 2024. [Online; accessed 3-June-2025]. Document ID: UG1366.

D. I. C. Silvano et al. A survey on deep learning hardware accelerators for heterogeneous hpc platforms. *ACM Computing Surveys*, 57(11):286:1–286:39, 2025. URL *https://dl.acm.org/doi/10.1145/3729215*.

W. Li and M. Liewig. A survey of ai accelerators for edge environment. *https://doi.org/10.1007/978-3-030-45691-7_4*, 2020.

S. I. Venieris et al. Toolflows for mapping convolutional neural networks on fpgas: A survey and future directions. *ACM Computing Surveys*, 51(3), jun 2018. URL *https://doi.org/10.1145/3186332*.