TRƯỜNG ĐẠI HỌC
SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH
HCMC University of Technology and Education

IMPLEMENTATION OF IMAGE SUPER RESOLUTION NETWORK BASED ON THE XILINX ZYNQ ULTRASCALE+ MPSOC – ZCU102 EVALUATION KIT

Student Name: Ha Nguyen Minh Phuoc ID: 20161245

Abstract

The project focuses on applying deep learning networks to the problem of image super-resolution. Specifically, it involves an in-depth study of the operational principles and architecture of the RCAN (Residual Channel Attention Network) neural network model, followed by training and experimentation on the ZCU102 platform.

Problem Statement

In the era of digitalization and technological advancement, images play a crucial role in communication and information transmission. The quality of images not only affects user experience but also significantly impacts various fields such as healthcare, security, industry.

Problem Solving

The project requires implementing the RCAN super-resolution neural network model using the PyTorch library, train on Kaggle and deploy on ZCU102.

- Design Hardware Platform use IP Cores for communication between Zynq Ultrascale+ MPSoC and the Deep Learning Processor Unit.
- Construct a boot system integrating the IP Cores design to run Linux operating system for Xilinx Ultrascale+ MPSoC - ZCU102.
- Perform pre-processing of the RCAN super-resolution image algorithm and deploy it on Xilinx Ultrascale+ MPSoC - ZCU102.

Implementation Details

The RCAN network consists of the following main components: shallow feature extraction, deep feature extraction using the Residual in Residual (RIR) structure, an upscaling module, and a reconstruction part.

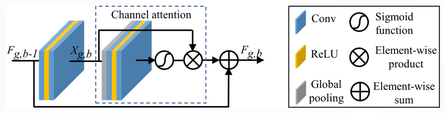


Figure 1: *Channel attention block*

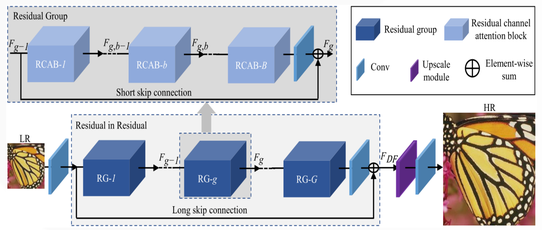


Figure 2: *Residual Channel Attention Networks*

The implementation flowchart of the system

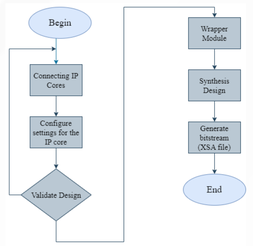


Figure 3: *Hardware platform design flow to generate XSA file in Vivado IDE*

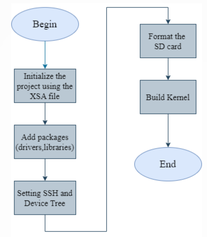


Figure 4: *Project design flow for Linux on PetaLinux tool*

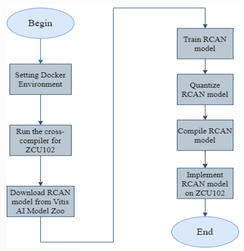


Figure 5: *System design flow for image super-resolution on Vitis AI tool*

Table 1: *The parameters of design hardware platform*

Number of DPU Core	1
Arch of DPU	B4096
Ram Usage	Low
Alu Relu Type	ReLU + ReLU6
Number of SMF cores	1
AXI Protocol	AXI4
clk_dsp	650 MHz
clk_dpu	325MHz
Duty Cycle	50%

Hardware connection

Establishing a data transmission connection between the PC and the ZCU102 through an Ethernet interface with an IPv4 address.

Writing the files built by PetaLinux tool onto the SD card for booting the ZCU102.

Set the power supply to 12V DC, and the UART speed to 115200., etc.

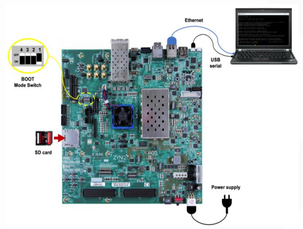


Figure 6: Hardware connection

Achieved Results

Design Hardware Platform

The IP core connectivity diagram of the system for the deep learning network on ZCU102 includes key IPs such as Zynq Ultrascale+ MPSoC, Deep Learning Processor Unit (DPU), Clock Wizard, and Processor System Reset.

Table 2: *Resources of IP cores when synthesys in Vivado*

Constrain	LUT	FF	BRAM	URAM	DSP
DPU_zynq_ultra_ps_e_0	264	0	0	0	0
DPU_clk_wiz_0_0	1	0	0	0	0
DPU_proc_sys_reset_0_0	19	40	0	0	0
DPU_proc_sys_reset_0_1	24	40	0	0	0
DPU_dpuczdx8g_0_0	61893	107918	259	0	724

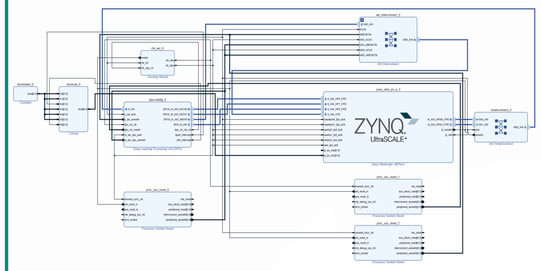


Figure 7: *IP connectivity diagram of the system*

The result of building the Linux kernel image, consists of boot files (such as boot.src, BOOT.BIN, ...) and configuration files (like pxelinux.cfg, rootfs.tar, ...).

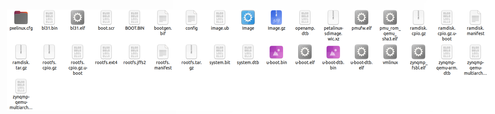
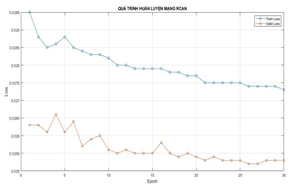


Figure 8: *Files generated during kernel build*

Pre-processing for the super-resolution network RCAN

The training data for the model is sourced from the DIV2K dataset, which consists of 800 high resolution images divided into training and test.



The loss values decrease gradually with each epoch.

Figure 9: *Loss value of RCAN during training on Kaggle*

Deploy RCAN onto ZCU102

Figure 10: *Results of super-resolution for the image butterfly.png from the Set5 dataset*

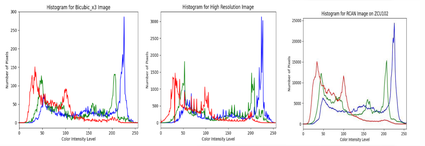


Figure 11: *Histogram chart of the image butterfly.png from the Set5 dataset*

Figure 12: *Bar chart of PSNR and SSIM values when running RCAN on Kaggle and ZCU102*

Conclusion & Future Direction

Conclusion

- Successfully establishing a hardware platform using IP cores.
- Building the Linux operating system.
- Deploying the RCAN super-resolution network to run on ZCU102.

Future Directions

Optimizing the model by adjusting the output image size.

Implementing other super-resolution networks such as VDSR, GAN, etc.

Reference

G. K. Ravi, "Deploying Deep learning Image Super-Resolution Models in Xilinx Zynq MPSoC ZCU102," 2020.

Xilinx, ZCU102 Evaluation Board User Guide (Ug1182) (v.17), 2023.

Xilinx, DPUCZDX8G for Zynq UltraScale+ MPSoCs - PG338 (v4.1), 2023.