

SuperRes

WHAT IS SUPERRES

- Super resolution imaging is a technique that enhances the resolution of an image or video.
- In addition to enhancing image quality, we want to minimize the execution time of the upscaling algorithm, this is where FPGA comes into play.
- Since FPGAs are good for intensive data computation and static parallel tasks, it provides a faster approach for image processing.

METHODOLOGY

- Identify the bottlenecks of the SuperRes application by running a profiler on the python application using TensorFlow.
- Design hardware IPs based on the bottleneck which is the whole ESPCN convolutional neural network.
- Use AXI Stream to transfer data between FPGA and CPU.
- Optimise the accelerator by quantization, this hugely reduce memory requirement and computational cost of using the network on hardware.

ALGORITHM FOR SUPERRES

- The ESPCN network has 4 hidden layers, the output is the same size as the input for each layer. Therefore, zero padding is required.
- ESPCN increases the resolution of the image at the last layer of the neural network, this reduces the computational complexity.
- The last layer is the efficient sub-pixel convolutional layer to recover the output image size with a specified upscale factor.

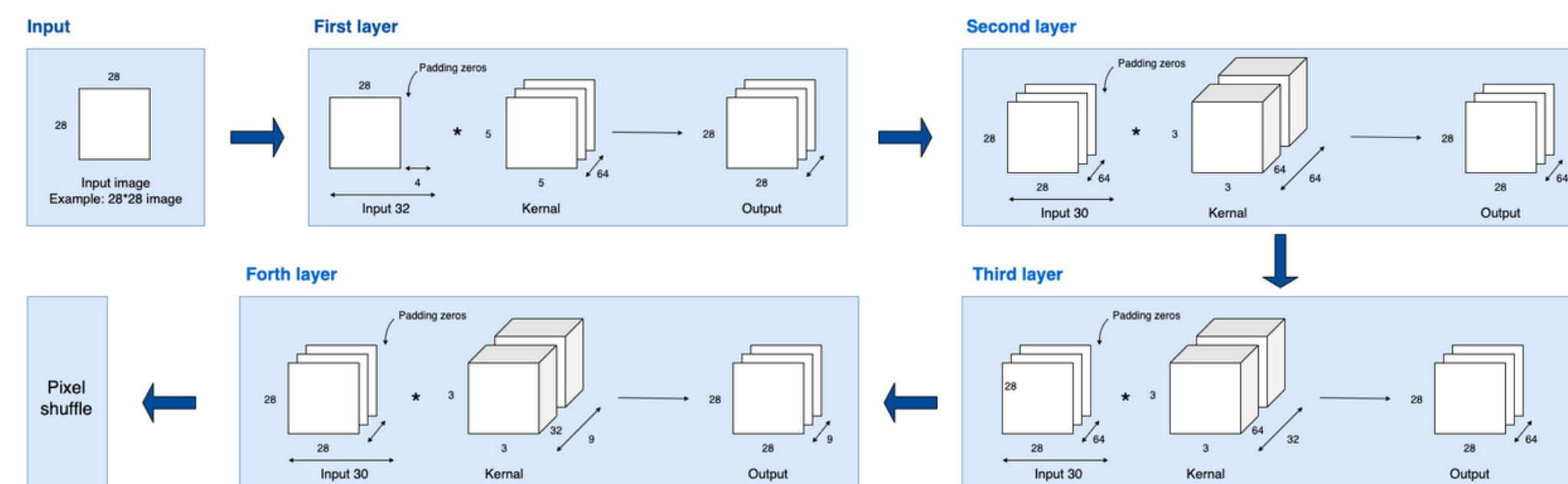


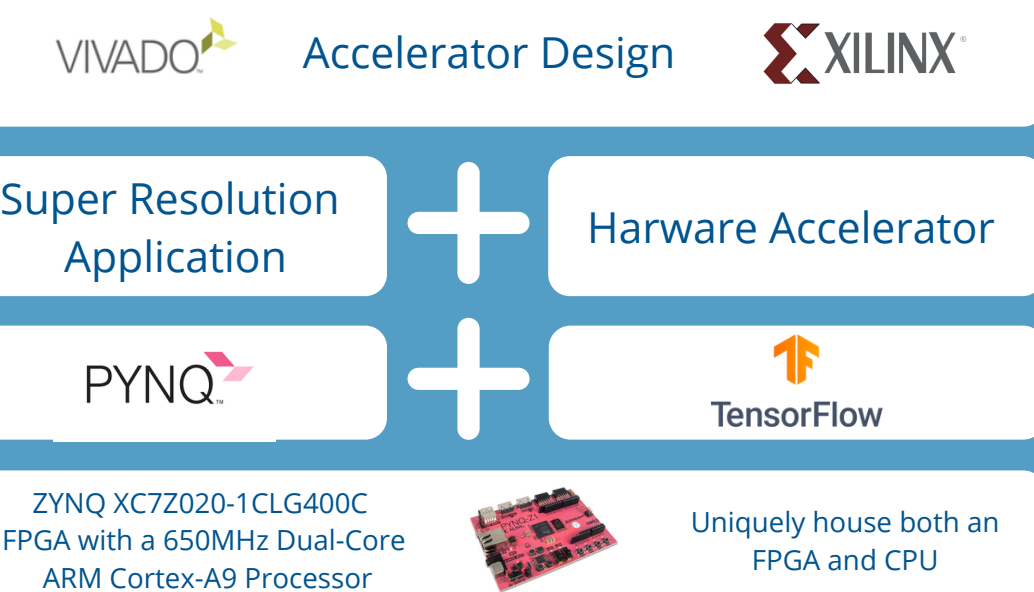
Figure 1: ESPCN Neural Network Architecture

ACKNOWLEDGEMENTS

We would like to thank:

- Manohar Vohra & Aswin Vivekanandan from ARM for providing innovative ideas in the data transfer between FPGA and CPU.
- Dr Bouganis for providing valuable insight into hardware IP design.
- Dr Wright for organizing the project.

FRAMEWORK



FUTURE WORK

- There are two types of quantization: quantization aware training and post training quantization. Compare the performance giving by two types quantization and choose the one with better accuracy.
- Using the pipeline directive to speed up the computation in hardware.
- Compare the performance with FINN - End-to-End Deep-Learning Framework for Fast Exploration of Quantized Neural Networks.
- Investigate the performance of the application by using different activation functions.

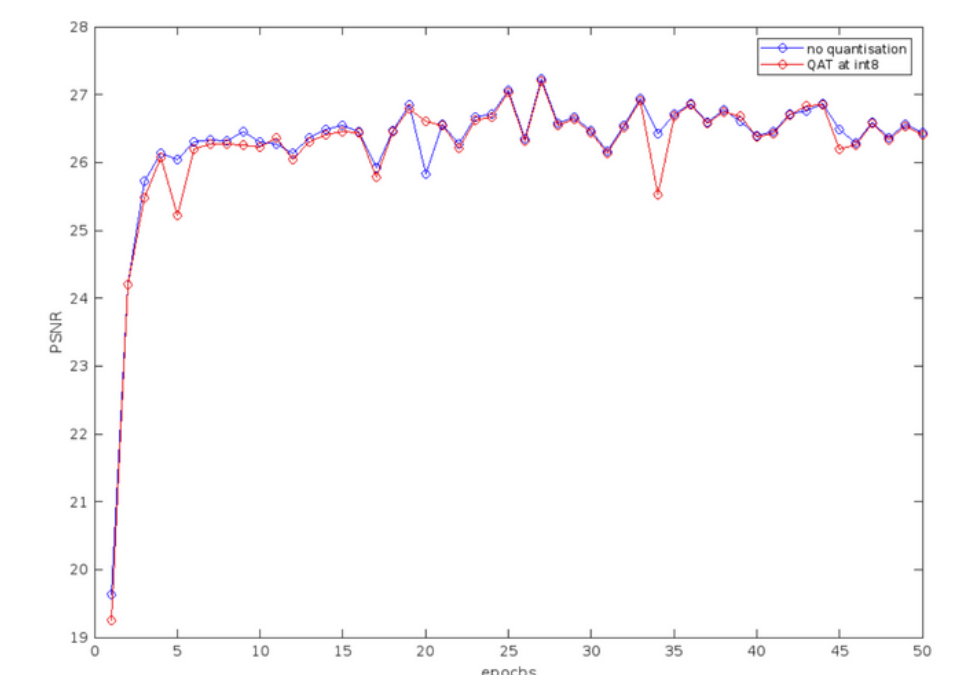


Figure 2: Model Performance