Utilizing SDSoC to Port Convolutional Neural Network to a Space-grade FPGA

SOUTHWEST RESEARCH INSTITUTE®

Josh Anderson
joshua.anderson@swri.org
Southwest Research Institute

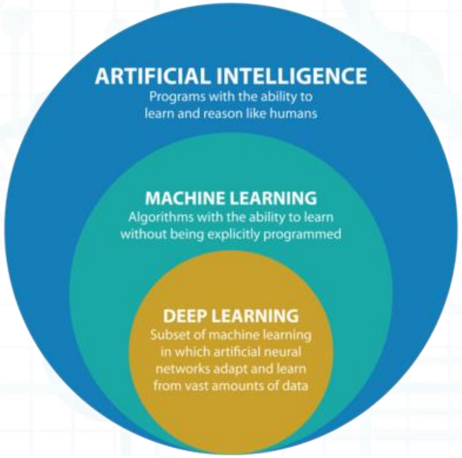
This is a non-ITAR presentation, for public release and reproduction from FSW website.

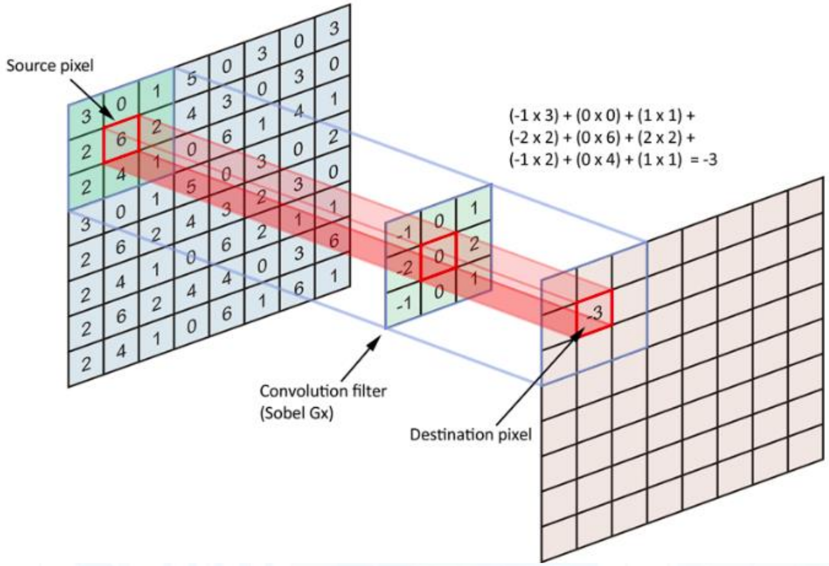
Objective

- Compress MASPEX instrument data
  - Produces ~80MB / sec!
- Port a Convolutional Neural Network (CNN) created for data compression onto an FPGA
  - Downlink only relevant information
- Compare utilization results to a typical space-grade FPGA

Convolutional Neural Network SparkNotes

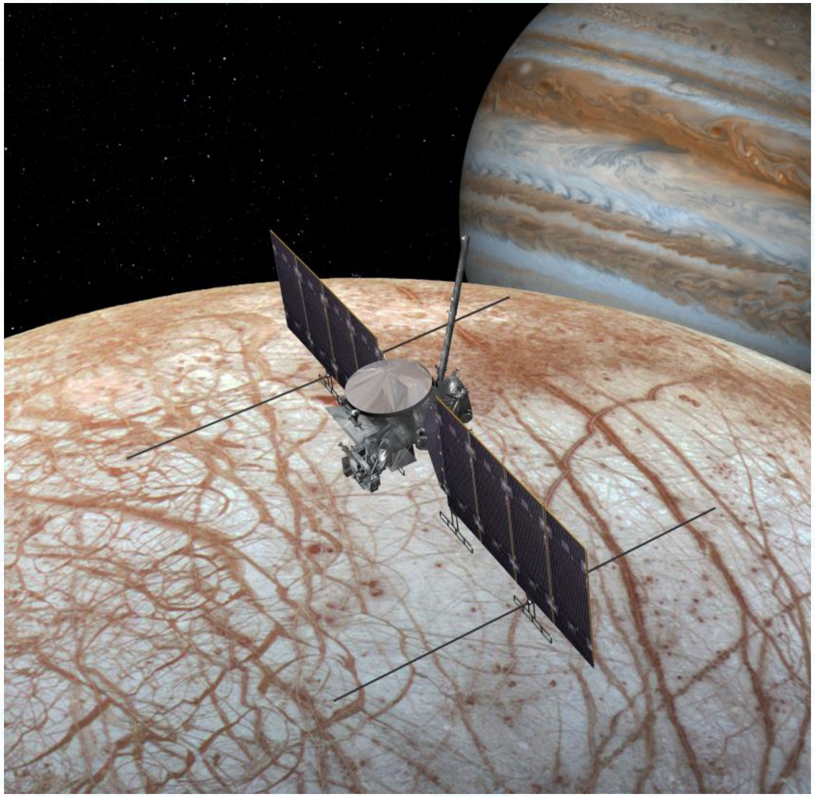
- A deep learning architecture
- Come in many shapes and sizes
- Applications include image recognition, object detection, and signal processing
- Primary operation is convolution on matrices


ARTIFICIAL INTELLIGENCE
Programs with the ability to learn and reason like humans
MACHINE LEARNING
Algorithms with the ability to learn without being explicitly programmed
DEEP LEARNING
Subset of machine learning in which artificial neural networks adapt and learn from vast amounts of data


Source pixel
(-1 x 3) + (0 x 0) + (1 x 1) +
(-2 x 2) + (0 x 6) + (2 x 2) +
(-1 x 2) + (0 x 4) + (1 x 1) = -3
-1 0 1
-2 0 2
-1 0 1
-3
Convolution filter (Sobel Gx)
Destination pixel

Why Machine Learning on FPGA?

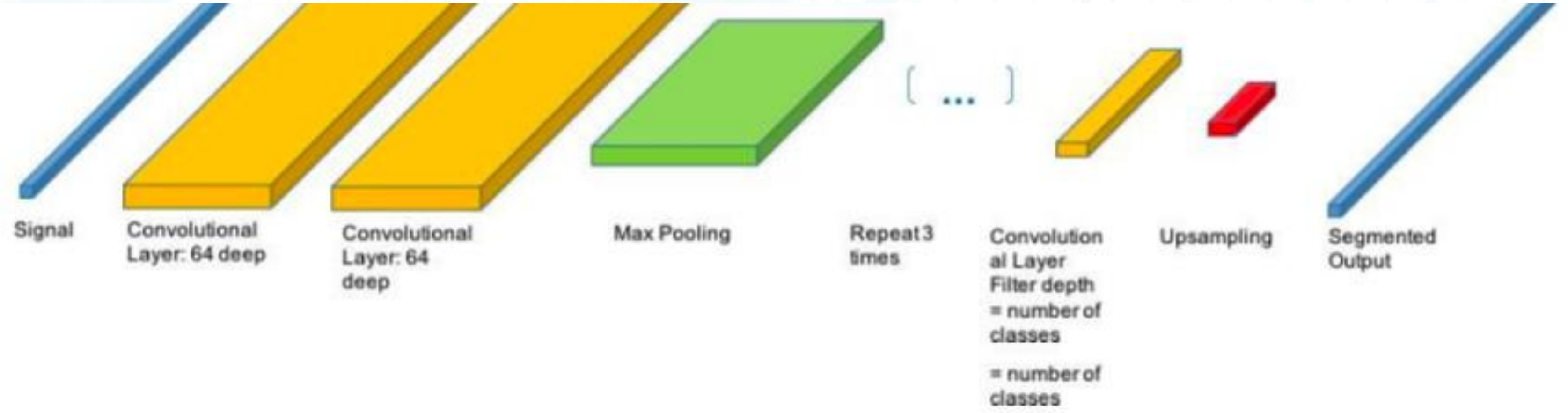
- Massive parallel computing power = Faster processing of data
- Low power
- Onboard processing
- FPGAs commonly used in space



This is a non-ITAR presentation, for public release and reproduction from FSW website.

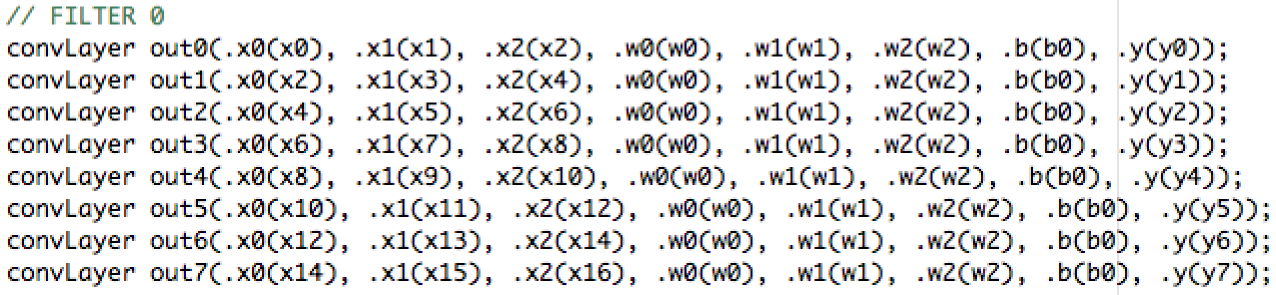
Network Architecture

- Optimized for reduced size
- 1D squeeze-net consisting of "Fire modules"
- Weights = 1.5 Mb to 236 Kb


[...]
Signal
Convolutional Layer: 64 deep
Convolutional Layer: 64 deep
Max Pooling
Repeat 3 times
Convolution al Layer Filter depth = number of classes
= number of classes
Upsampling
Segmented Output

Hardware Implementation – Simple Approach

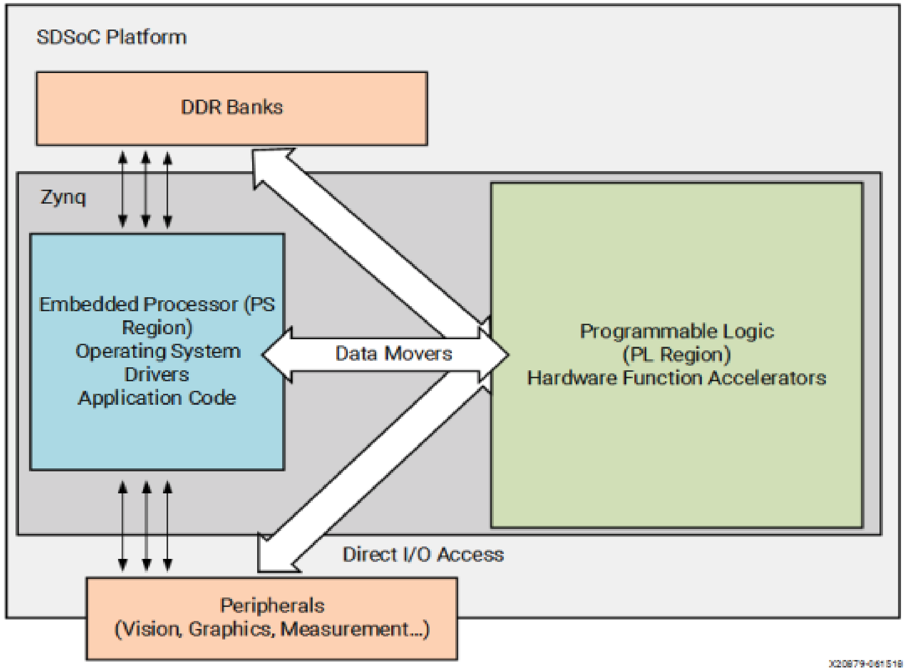
- Brute force = ineffective
- Prototype small network on Virtex-5 with Verilog
  - 1 small convolutional layer
    - 32 16-bit inputs and 16 filters
    - **Utilization = 108%**
- Fixed point still not enough
- Final network has over 20 convolutional layers!



```
// FILTER 0
convLayer out0(.x0(x0), .x1(x1), .x2(x2), .w0(w0), .w1(w1), .w2(w2), .b(b0), .y(y0));
convLayer out1(.x0(x2), .x1(x3), .x2(x4), .w0(w0), .w1(w1), .w2(w2), .b(b0), .y(y1));
convLayer out2(.x0(x4), .x1(x5), .x2(x6), .w0(w0), .w1(w1), .w2(w2), .b(b0), .y(y2));
convLayer out3(.x0(x6), .x1(x7), .x2(x8), .w0(w0), .w1(w1), .w2(w2), .b(b0), .y(y3));
convLayer out4(.x0(x8), .x1(x9), .x2(x10), .w0(w0), .w1(w1), .w2(w2), .b(b0), .y(y4));
convLayer out5(.x0(x10), .x1(x11), .x2(x12), .w0(w0), .w1(w1), .w2(w2), .b(b0), .y(y5));
convLayer out6(.x0(x12), .x1(x13), .x2(x14), .w0(w0), .w1(w1), .w2(w2), .b(b0), .y(y6));
convLayer out7(.x0(x14), .x1(x15), .x2(x16), .w0(w0), .w1(w1), .w2(w2), .b(b0), .y(y7));
```

Solution – SDSoC Software Defined System on a Chip

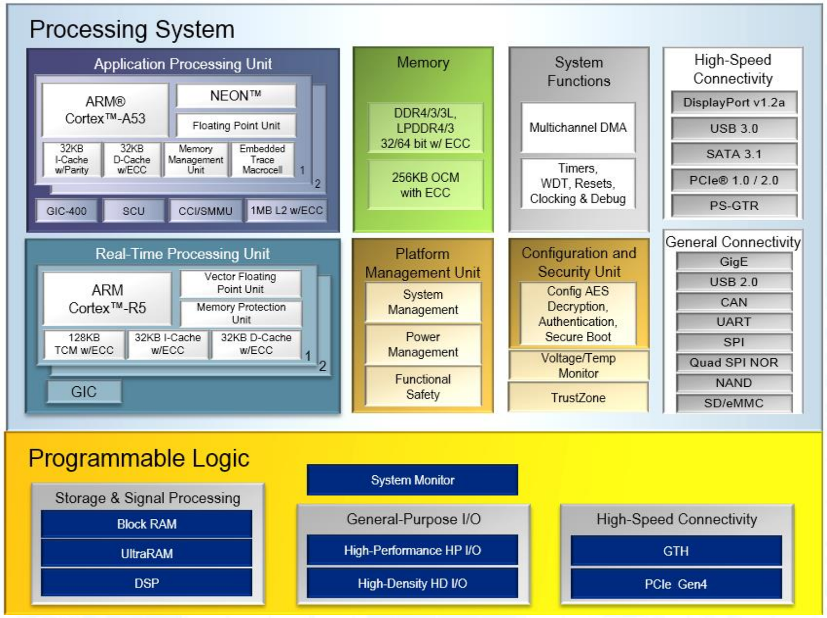
- Write applications for Multi-Processor System on a Chip (MPSoC) devices
- Synthesizes C/C++ code onto programmable logic (PL) of MPSoC
- Programmable System (PS)
- Programmable Logic (PL)
- Orchestrates moving data between PS and PL

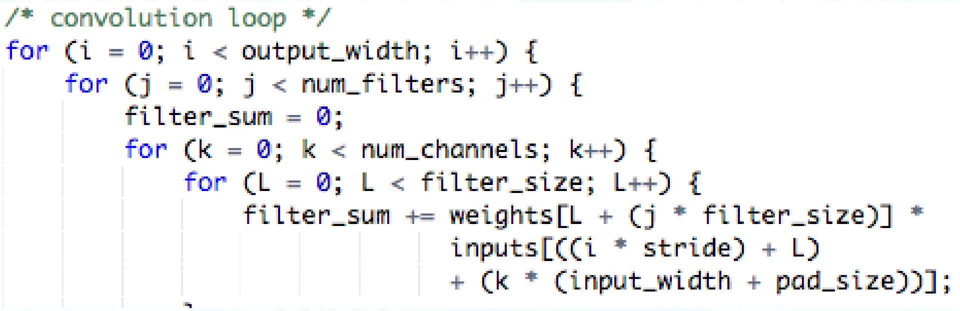


This is a non-ITAR presentation, for public release and reproduction from FSW website.

Hardware Implementation – A Better Approach

- Zynq Ultrascale+ MPSoC
- Model as close to a stand-alone FPGA as possible
- Reusable convolutional layer written in C++ synthesized onto PL





```
/* convolution loop */
for (i = 0; i < output_width; i++) {
    for (j = 0; j < num_filters; j++) {
        filter_sum = 0;
        for (k = 0; k < num_channels; k++) {
            for (L = 0; L < filter_size; L++) {
                filter_sum += weights[L + (j * filter_size)] *
                                  inputs[((i * stride) + L)
                                  + (k * (input_width + pad_size))];
```

Utilization Results

- Close to 0 utilization of DSP48E, FF, and LUT cells
- Weights/biases stored in BRAM

Full-size 1DCNN on Zynq UltraScale+

Name	BRAM_18K	DSP48E	FF	LUT
Total	706	16	3616	12532
Available	1824	2520	548160	274080
Utilization (%)	**39**	**~0**	**~0**	**4**

This is a non-ITAR presentation, for public release and reproduction from FSW website.

Utilization Results cont.

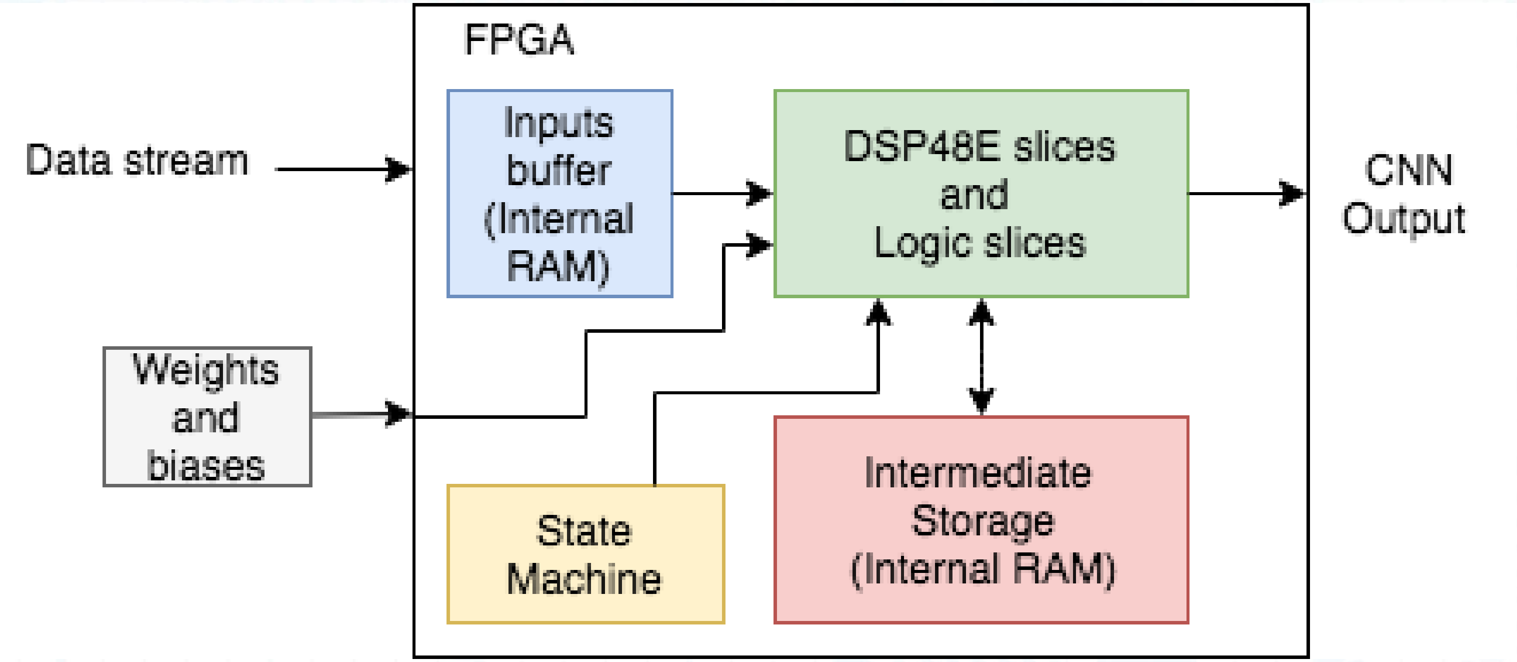
- 2D network architecture
- Useful for image processing

2DCNN Experimentation

Name	BRAM_18K	DSP48E	FF	LUT
Total	706	28	5046	13927
Available	1824	2520	548160	274080
Utilization (%)	**39**	**1**	**~0**	**5**

Hypothetical FPGA Design

- State machine logic replaces embedded processor
- Store weights and biases externally


FPGA
Data stream
Inputs buffer (Internal RAM)
DSP48E slices and Logic slices
CNN Output
Weights and biases
State Machine
Intermediate Storage (Internal RAM)

This is a non-ITAR presentation, for public release and reproduction from FSW website.

Moving to Space-grade FPGAs

- Rough estimation
- Assumptions:
  - Similar results on DSP48E and LUTs
  - External storage of weights/biases
  - State machine logic not accounted for

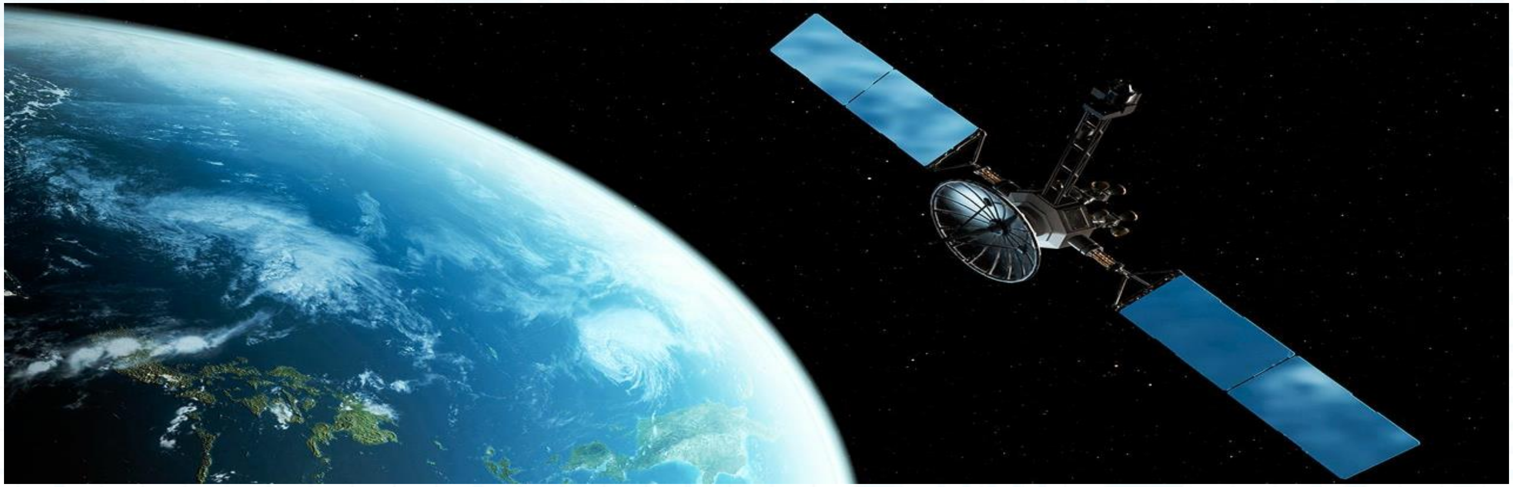
Extrapolating to Virtex-5QV

Name	BRAM_18K	DSP48E	FF	LUT
Total	706	16	3616	12532
Available	596	320	40960	40960
Utilization (%)	**118**	**5**	**8.83**	**30.60**

This is a non-ITAR presentation, for public release and reproduction from FSW website.

Conclusion

- A CNN for data compression could potentially be implemented on a space-grade FPGA
- Network could be used to compress data onboard
  - Reduce required downlink volume
- SDSoC can be used as a tool to prototype and benchmark design
  - Optimize before hardware description language (HDL) implementation



What's next?

- Move to non-MPSoC implementation (i.e. Virtex 5 FPGA)
- Compare speed and scalability vs GPU
- Improve resource utilization
- Explore higher dimensional data, such as 2D EO/IR imaging and video

This is a non-ITAR presentation, for public release and reproduction from FSW website.

Questions?

Thanks!

This is a non-ITAR presentation, for public release and reproduction from FSW website.

Sources

- https://www.jpl.nasa.gov/missions/web/europa_full.jpg
- https://www.xilinx.com/support/documentation/data_sheets/ds192_V5QV_Device_Overview.pdf
- https://www.xilinx.com/support/documentation/data_sheets/ds891-zynq-ultrascale-plus-overview.pdf
- https://www.xilinx.com/content/dam/xilinx/imgs/products/zynq/zynq-cg-block.PNG
- https://www.xilinx.com/support/documentation/sw_manuals/xilinx2018_2/ug1027-sdsoc-user-guide.pdf
- https://www.qubole.com/blog/deep-learning-the-latest-trend-in-ai-and-ml/
- https://medium.freecodecamp.org/an-intuitive-guide-to-convolutional-neural-networks-260c2de0a050
- https://www.nbcnews.com/mach/video/from-the-cold-war-to-hurricanes-the-evolution-of-space-satellites-1097831491829?v=railb&