Machine Learning Heterogeneous Computing

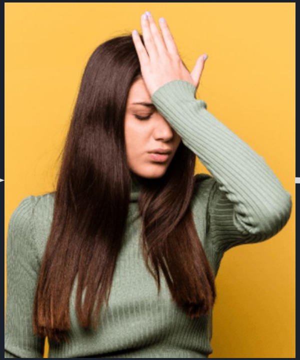
Members: Alek Comstock, Jeffery Kasper, Sandro Panchame, Rudy Nahra

Advisor: Dr. Rover
Client: JR Spidell

Project Plan

Problem Statement

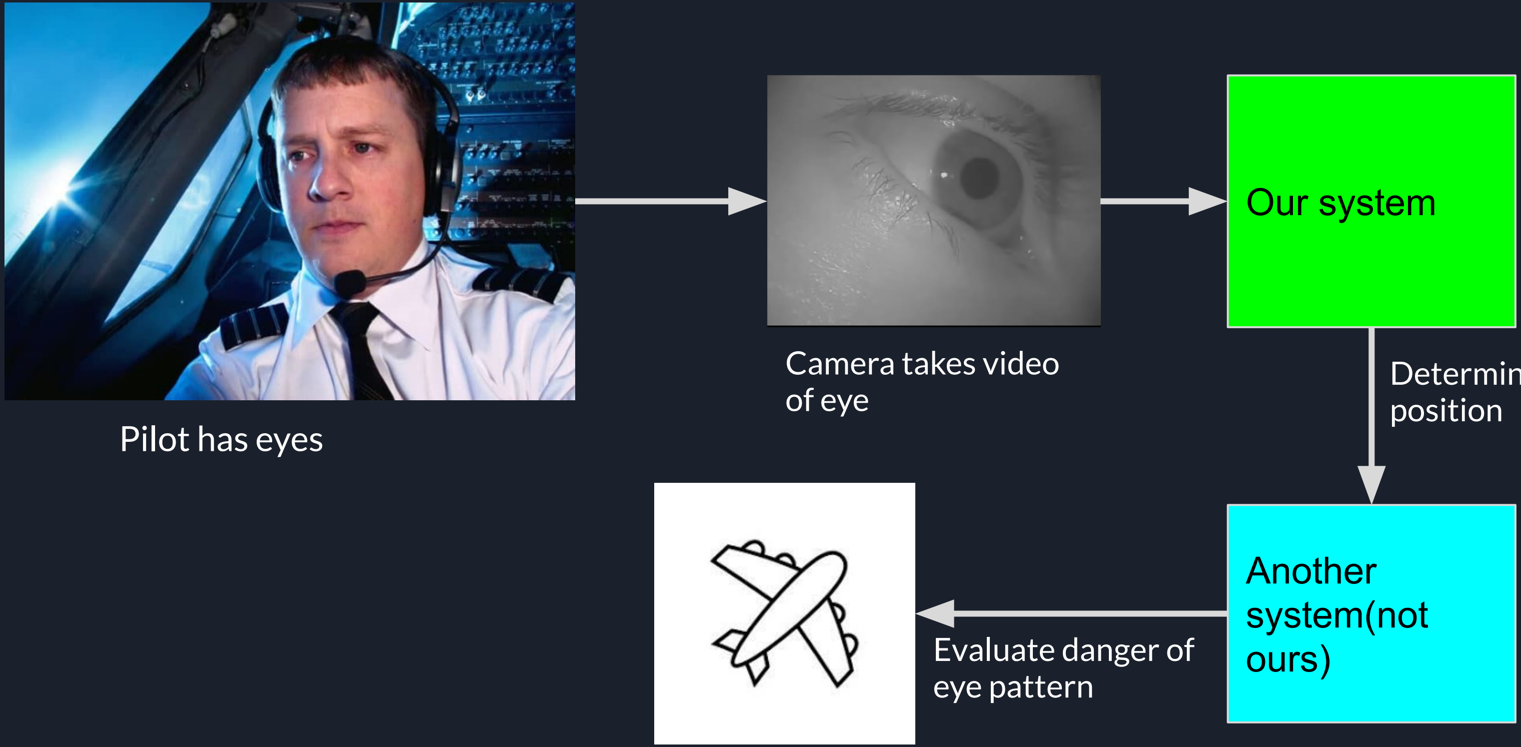
Pilot - extreme stress



Bad decision making

Danger

Solution Concept



Pilot has eyes

Camera takes video of eye

Our system

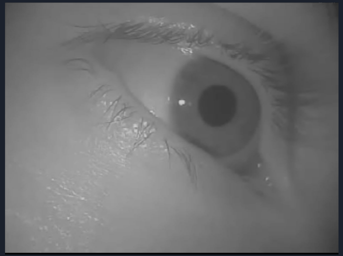
Determine eye position

Another system(not ours)

Evaluate danger of eye pattern

Lock pilot controls, use autopilot

Requirements



Functional Requirements

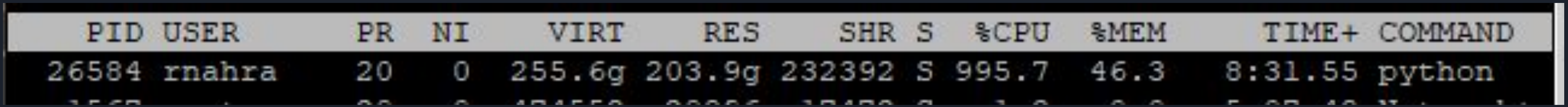
- System takes in a video feed or someone’s eye and for each frame, outputs:
  - Boolean value for blinking or not blinking
  - (x,y) integers of the position of the pupil in the image

Non-functional Requirements

- System must be able to process frames with higher throughput than they are received
- Design should be able to adapt to use all available cores
- Deep Learning model will be verified to be safe using Marabou, the neural network verification tool

Technical/Other Constraints/Considerations

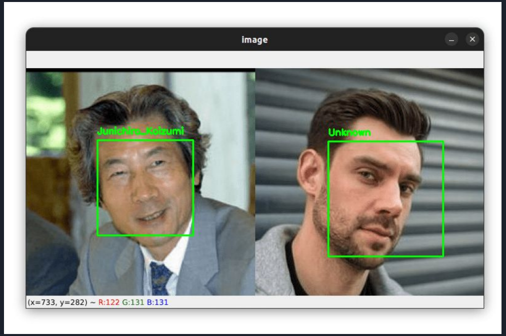
- Two Deep Learning Models
  - Classifier model - blinking or not
  - Regression model - find position of pupil
- Xilinx Kria SOM Board
  - FPGA Board with 4GB RAM, 4 APU Cores, 2 RPU cores
  - DPU Hardware Accelerator in the programmable logic
- NP-completeness of neural network verification problem



```
  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
26584 rnahra    20   0  255.6g 203.9g 232392 S 995.7  46.3   8:31.55 python
```

Market survey

- Similar Devices on the Market
  - Self-Driving Cars
  - Face Recognition
  - Eye Tracking
  - Find Disease
- Our project:
  - Specialized hardware on board
  - Board setup to process continually and efficiently
  - Low cost - other systems roughly $10k+



(x=733, y=282) ~ R:122 G:131 B:131

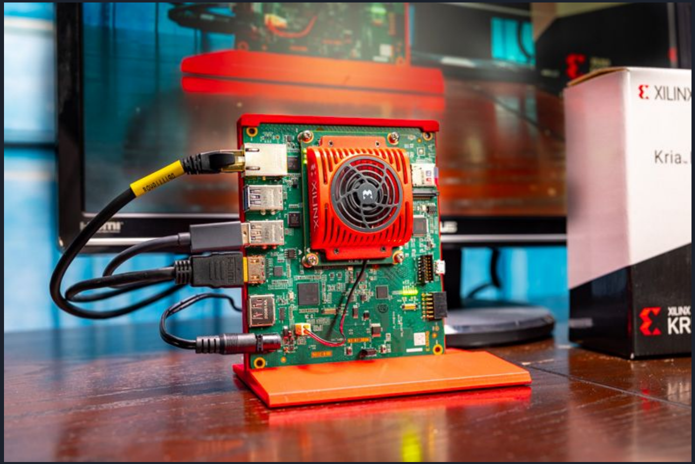
Potential Risks & Mitigation

- NP-completeness of Neural Network Verification/Marabou [Overall Risk Factor: 1]
    - Mitigation Plan: Make compromises on size of neural network
- Computation time [Overall Risk Factor: 0.8]
    - Mitigation Plan: We have budgeted the ability to purchase DPU fabric expansion cards. We will need to setup and program these cards.

Resource/Cost Estimate

- Kria SOM kv260 board
  - SD cards
  - Power Cables
  - USB-b mini data cable
- Camera

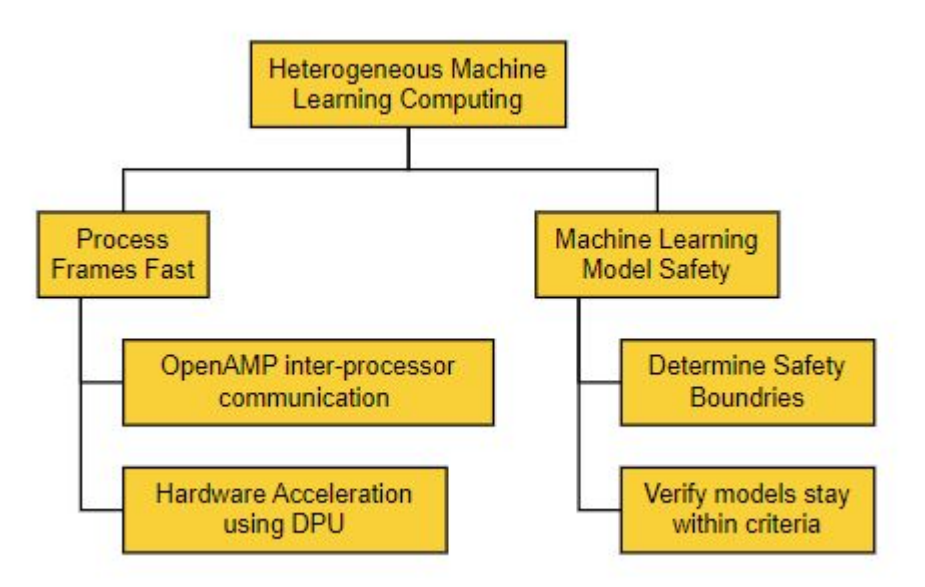
All resources are being supplied by the client.



System Design

Functional Decomposition

- Broken into two parts
- Machine learning optimization
  - Maraboupy
  - Two models
  - Meeting safety criteria
- Hardware and Application
  - Use hardware to speed up processing time
  - Use OpenAMP to distribute work



Heterogeneous Machine Learning Computing

Process Frames Fast

OpenAMP inter-processor communication

Hardware Acceleration using DPU

Machine Learning Model Safety

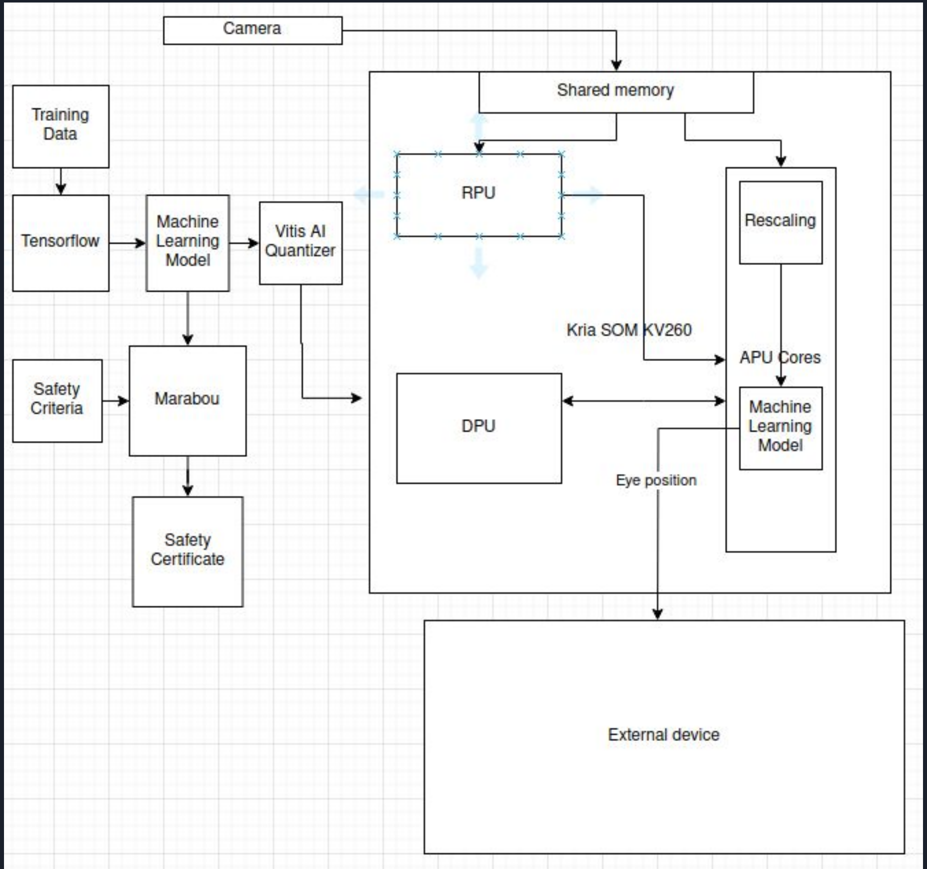
Determine Safety Boundries

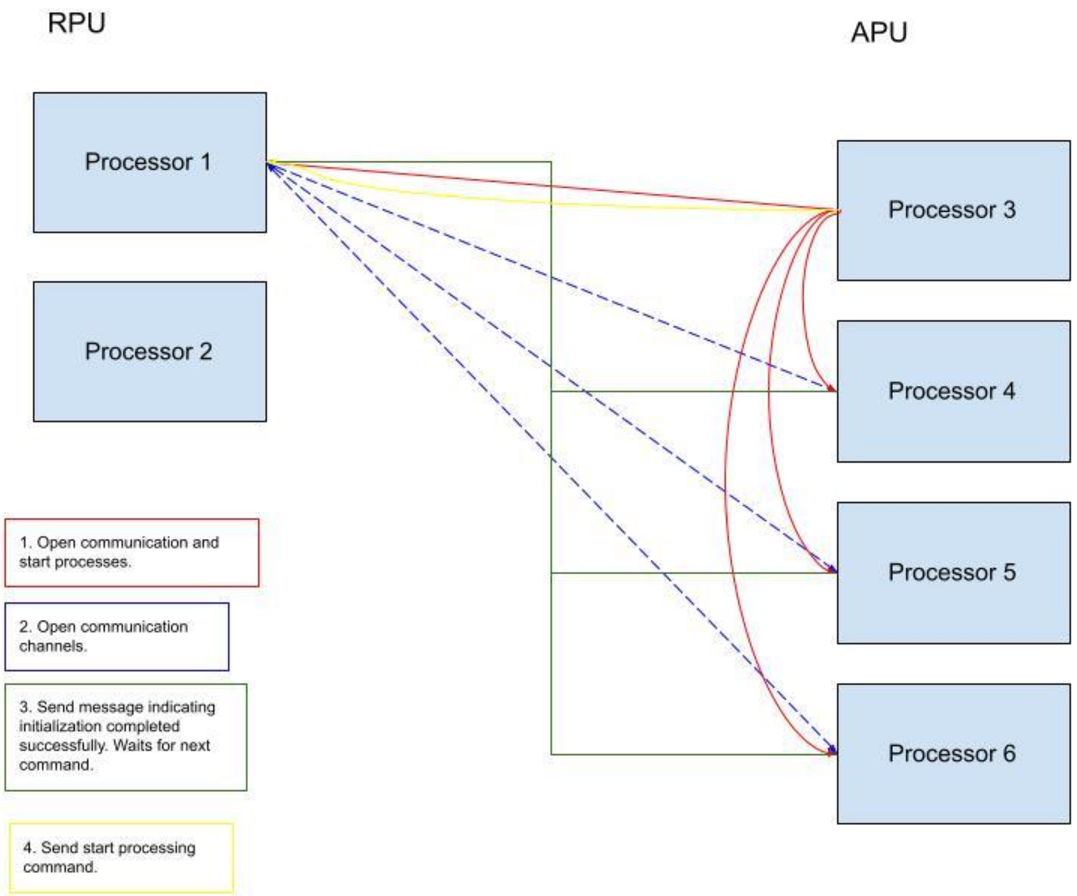
Verify models stay within criteria

Detailed Design

Hardware:

- Split up memory into 4 1 GB segments
- Each worker CPU (3 of APUs) manages segment of memory and DPU
- Distributor core (1 of RPU) uses last memory segment to manage workers
- Worker CPUs manage video frames for DPU





1. Open communication and start processes.

2. Open communication channels.

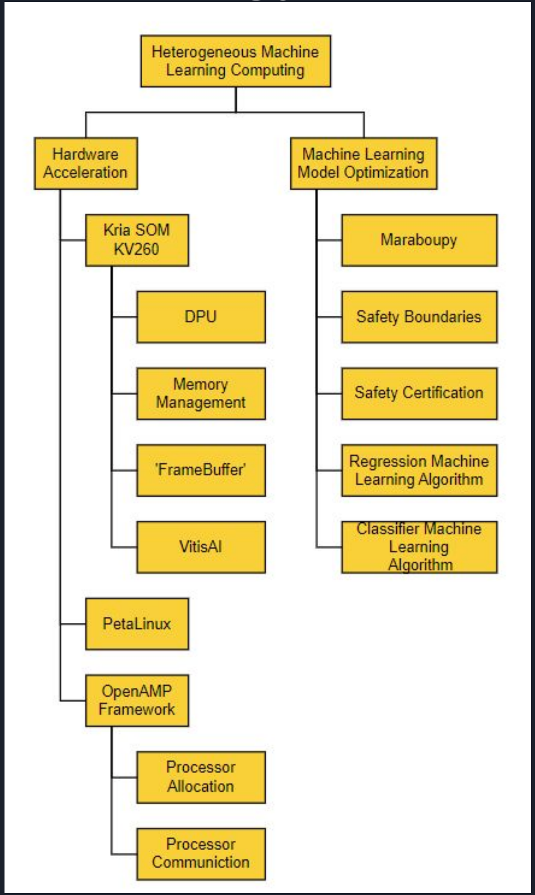
3. Send message indicating initialization completed successfully. Waits for next command.

4. Send start processing command.

HW/SW/Technology Platform(s) used

Hardware

- PetaLinux
- openAMP
- VitisAI



- Heterogeneous Machine Learning Computing
  - Hardware Acceleration
    - Kria SOM KV260
      - DPU
      - Memory Management
      - 'FrameBuffer'
      - VitisAI
    - PetaLinux
    - OpenAMP Framework
      - Processor Allocation
      - Processor Communiction
  - Machine Learning Model Optimization
    - Maraboupy
    - Safety Boundaries
    - Safety Certification
    - Regression Machine Learning Algorithm
    - Classifier Machine Learning Algorithm

Software

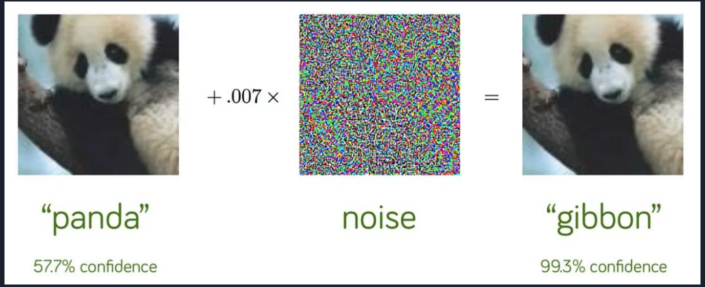
- Tensorflow
- Python and many libraries (opencv, pandas, etc)
- Marabou - Neural Network Verification tool

Test Plan

- Custom message passing between processors at runtime
  - Message should originate in ‘Controller’ APU thread and is passed to RPU ‘Distributor’ thread.
  - Message is sent to all ‘Worker’ APU threads and worker threads reply to the message.
  - Distributor thread receives replies and forwards them to the controller thread.
  - Controller thread logs the replies.

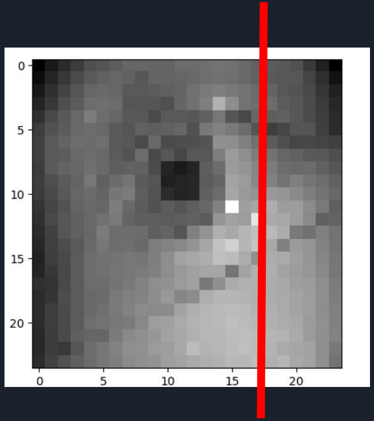
Testing - Neural Network & Marabou

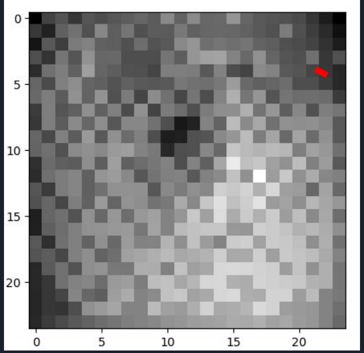
- How do you test a neural network?
  - Standard evaluation and calculation of error, accuracy, etc.
  - Not informative enough - Is it *too* wrong? Or just wrong enough?
- Formal verification with Marabou
  - Marabou can determine if some output can be reached given some constraints on the input
  - If some small change to an eye can cause an unacceptable result, network is unsafe
  - Marabou will guarantee that our model is "safe" given our safety criteria



Prototype Implementations or basic building block implementations

- Custom message passing
- Simple Marabou NN verification





Delta=0.05

Delta=0.03: unsat

Concluding Thoughts

Milestones & Project Status

Completed	In Progress	Future
Install Petalinux & OpenAMP on Kria SOM board	Get Inter-Process Communications working	Get Machine Learning Algorithms onto the board using Vitis
Research into Machine Learning	Develop safety criteria with Marabou	Train Neural Network on larger dataset
Basic verification queries with Marabou	Determine feasible network size for Marabou	Ensure processing speed is 60fps on average
Native OpenAMP Echo test		Tune and modify neural network until it passes safety tests

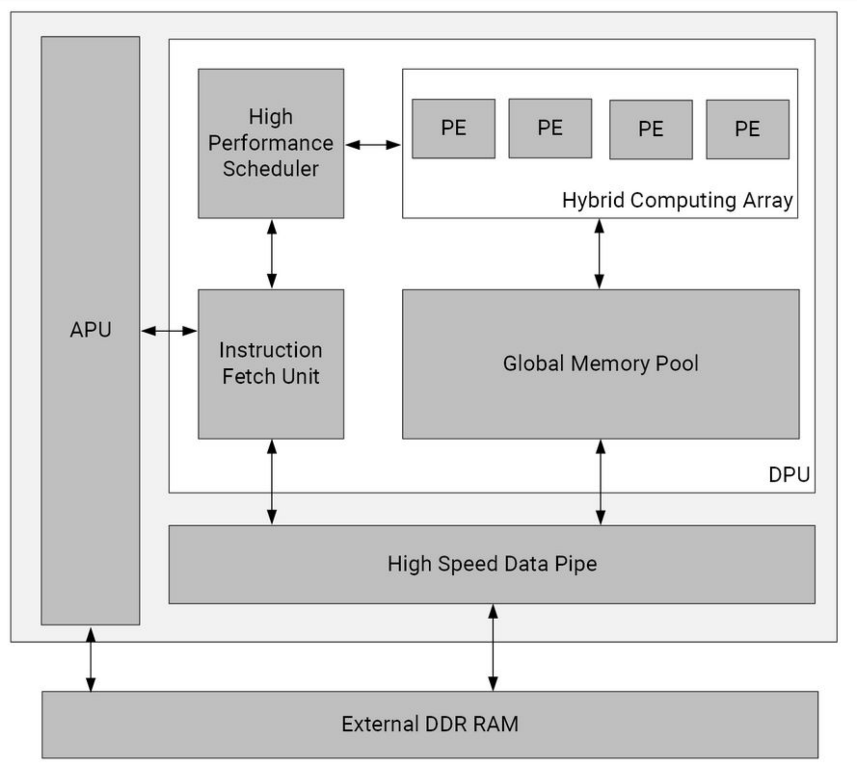
Plan for next semester

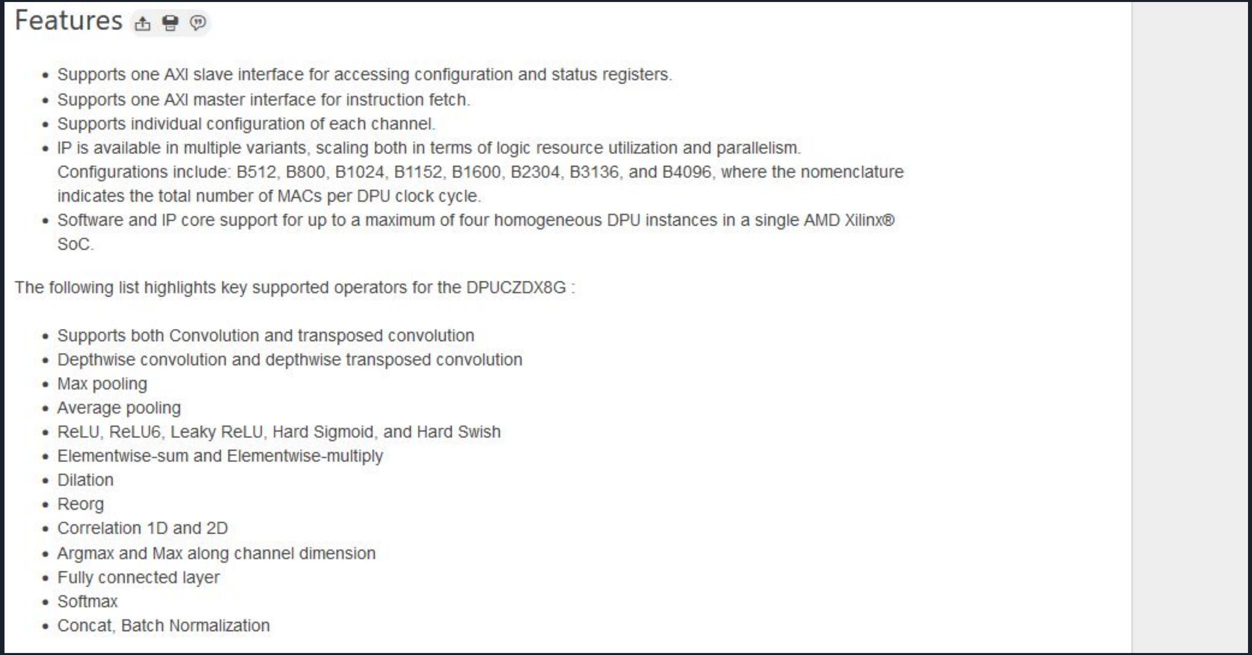
- Get machine learning algorithms onto the board using VitisAI
- Train NN on larger dataset
  - The current dataset is small, some of the data works.
- Ensure processing speed is 60fps on avg.
  - Running assumptions on processing times
- Tune and modify NN until it passes safety tests

Appendix

Task responsibility/contributions of each project member

- Rudolph Nahra
  - Neural Network Analysis and Optimization
- Sandro Panchame
  - Neural Network Analysis and Optimization
- Jeffrey Kasper
  - Embedded Systems Design
  - Operating Environment Developer
- Alek Comstock
  - Embedded Systems Design


High Performance Scheduler
PE
PE
PE
PE
Hybrid Computing Array
APU
Instruction Fetch Unit
Global Memory Pool
DPU
High Speed Data Pipe
External DDR RAM



Features

- Supports one AXI slave interface for accessing configuration and status registers.
- Supports one AXI master interface for instruction fetch.
- Supports individual configuration of each channel.
- IP is available in multiple variants, scaling both in terms of logic resource utilization and parallelism. Configurations include: B512, B800, B1024, B1152, B1600, B2304, B3136, and B4096, where the nomenclature indicates the total number of MACs per DPU clock cycle.
- Software and IP core support for up to a maximum of four homogeneous DPU instances in a single AMD Xilinx® SoC.

The following list highlights key supported operators for the DPUCZDX8G :

- Supports both Convolution and transposed convolution
- Depthwise convolution and depthwise transposed convolution
- Max pooling
- Average pooling
- ReLU, ReLU6, Leaky ReLU, Hard Sigmoid, and Hard Swish
- Elementwise-sum and Elementwise-multiply
- Dilation
- Reorg
- Correlation 1D and 2D
- Argmax and Max along channel dimension
- Fully connected layer
- Softmax
- Concat, Batch Normalization

Engineering Standards

https://ieeexplore.ieee.org/document/9726144

7001-2021 - IEEE Standard for Transparency of Autonomous Systems.

This standard is important for our project because we need to analyze our deep learning model to ensure its output is safe, as our project could be employed in safety-critical applications. The standard will help us measure how safe they are.

https://standards.ieee.org/ieee/29119-2/7498/

29119-2-2021 - Iso/iec/ieee international standard - software and systems engineering - software testing -- part 2: test processes - redline

This standard is useful to use as a form of ensuring we meet many possible problems our project could run into. It ensures that each step along the way, any updates or changes to the project, should meet our testing standards.

https://ieeexplore.ieee.org/document/1176958

1532-2002 - IEEE Standard for In-System Configuration of Programmable Devices

This standard applies to our project because we will utilize an FPGA board. An FPGA board is a type of Programmable Device. We will need to utilize these standards to effectively configure the board.